

# The microtype package

Subliminal refinements towards typographical perfection

R Schlicht  
w.m.1@gmx.net

v2.7  
2017/07/07

---

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 © 2004–2017 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 [100]	
14.3	Configuration . . . . .	103
	Font sets [103] Variants and aliases [108] Disabling ligatures [110] Interaction with babel [110] Fine tuning [110] Character inheritance [117] Permuta- tion [119]	

14.4	Package options . . . . .	122
	Declaring the options [122] Loading the definition file [126] Reading the configuration file [126] Hook for other packages [127] Changing options later [128] Processing the options [131]	
<b>15</b>	<b>Configuration files</b>	<b>140</b>
15.1	Font sets . . . . .	140
15.2	Font variants and aliases . . . . .	141
15.3	Interaction with babel . . . . .	143
15.4	Note on admissible characters . . . . .	143
15.5	Character inheritance . . . . .	143
	OT1 [144] T1 [144] LY1 [145] OT4 [145] QX [146] T5 [146] EU1, EU2, TU [147] Euro symbols [148]	
15.6	Tracking . . . . .	148
15.7	Font expansion . . . . .	148
15.8	Character protrusion . . . . .	150
	Normal [151] Italics [159] Small caps [170] Italic small caps [173] Text companion [175] Computer Modern math [179] AMS symbols [183] Euler [187] Euro symbols [191]	
15.9	Interword spacing . . . . .	191
	Nonfrenchspacing [194]	
15.10	Additional kerning . . . . .	195
	French [195] Turkish [196]	
<b>16</b>	<b>OpenType configuration files</b>	<b>197</b>
16.1	Character inheritance . . . . .	197
16.2	Character protrusion . . . . .	201
<b>17</b>	<b>Auxiliary file for micro fine tuning</b>	<b>214</b>
<b>A</b>	<b>The title logo</b>	<b>216</b>
A.1	Macros . . . . .	216
A.2	Document . . . . .	221
<b>B</b>	<b>The letterspacing illustration</b>	<b>221</b>
B.1	Macros . . . . .	222
B.2	Document . . . . .	225
<b>C</b>	<b>Change history</b>	<b>226</b>
<b>D</b>	<b>Index</b>	<b>234</b>
<b>E</b>	<b>The L<sup>A</sup>T<sub>E</sub>X Project Public License</b>	<b>243</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, the `babel` package), which may not always be a robust solution. In contrast to the standard kerning built into the fonts (which will of course apply as usual), this additional kerning relates to single characters, not to character pairs.

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

The possibility, finally, to *disable all ligatures* in a font is particularly useful for typewriter fonts.

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

## 2 Getting started

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

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

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

### 3 Options

Like many other  $\LaTeX$  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 version pdfT<sub>E</sub>X 1.20 (and with 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 the instances 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, automatic expansion even works in DVI mode, however, because postprocessing programs like `dvips` or `dvipdfmx` are (at the moment of this writing) not 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 1em; 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 XeTeX, 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. Neither letterspacing nor *automatic* font expansion will work because the postprocessing drivers (`dviips`, `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. This package defines a font set called ‘`basictext`’ in the main configuration file 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 =  $\langle encoding \rangle / \langle family \rangle / \langle series \rangle / \langle shape \rangle / \langle size \rangle$ '. 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 pdfTeX, this is not recommended, however, since it entails that kerning will be switched off, too. With LuaTeX, there is no such limitation. The default settings disable ligatures for the character 'f' only, i.e., 'ff',

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

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

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

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

and then write:

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

this would be the (typographically dubious) outcome:



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 *dimension* and, additionally, *space*. The latter will measure the values in thousandths of the respective space dimension set by the font. By default, the unit is measured by the space dimensions. For example, with the following (nonsensical) settings:

```

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

```

the space inserted after a full stop would be doubled (technically speaking:  $2 \times \text{\fontdimen 2}$ ), as would the maximum stretch and shrink amounts of the interword space ( $\text{\fontdimen 3}$  and  $4$ ). 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 `\lstyle` 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, ‘*Ausſichtslosigkeit*’, with ligatures shown in red, inhibited ligatures in green).

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

```
\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{120}
\textfrak{\lstyle 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 large a 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, which uses a different technique of expansion, the file size increase 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 `pdfcprot`). 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 or LuaTeX 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 team, 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  $X_{\text{E}}\text{T}_{\text{E}}\text{X}$  and  $\text{LuaT}_{\text{E}}\text{X}$  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  $\text{LuaT}_{\text{E}}\text{X}$  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 Dyballa*, *Morten Høgholm*, *Steven Bath*, *Daniel Flipo*, *Michalis Miatidis*, *Sven Naumann*, *Ross Hetherington*, *Wiebke Petersen*, *Geoff Vallis*, *Steven E. Harris*, *Karl Berry*, *Peter Meier*, *Nathan Rosenblum*, *Wolfram Schaalo*, *Vasile Gaburici*, *Sveinung Heggen*, *Axel Berger*, *Colin Rourke*, *Maverick Woo*, *Silas S. Brown*, *Lars Rönnbäck*, *Christian Stark*, *Leo*, *Marcin Borkowski*, *hscm*, *George Gratzner*, *Josep Maria Font*, *Juan Acevedo*, *Heiko Oberdiek*, *Till A. Heilmann*, *Rolf Dieterich*, *Seamus Bradley*, *Meho R*, *Steffen Hoffmann*, *Scott Pakin*, *Maïeul Rouquette*, *Jonas Hogstrom*, *Gabriel Kerneis*, ‘RazorXsr’, *Sebastian Schubert*, ‘Dave’, *Giuseppe Palma*, *Stephan Stiller*, *Christopher Schramm*, ‘uli’, *Sam Mason*, ‘kleenstar’, ‘Henning’, *Ronnie Marks*, *David Carlisle*, ‘Max’, ‘HcN’, *Will Robertson*, ‘user11126’, *Ulrike Fischer*, ‘Daniel’, ‘lcomdata’, *Reinhard Kotucha*, and ‘jcr’.

## 12 References

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

Hàn Thế Thành, ‘Font-specific issues in  $\text{pdfT}_{\text{E}}\text{X}$ ’, in: *TUGBoat*, vol. 29 (2008), no. 1: ‘EuroBach $\text{T}_{\text{E}}\text{X}$  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 pdf $\text{T}_{\text{E}}\text{X}$  user manual*, 20 April 2017. (Available from CTAN at </systems/doc/pdf-tex/manual/pdf-tex-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, Khaled Hosny, *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*, 31 March 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]
- Hook `\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 L<sup>A</sup>T<sub>E</sub>X font loading
- New option: ‘final’ [3.5]

**1.2 (2004/10/03)**

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

**1.1 (2004/09/21)**

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

**1.0 (2004/09/11)**

- First CTAN release

## 14 Implementation

The `docstrip` modules in this file are:

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

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

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

`xetex-def`: Definitions specific to `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}}
\MT@info@nl    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@error      82 \def\MT@error{\PackageError\MT@MT}
\MT@warn@err   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@info      0: almost none
\MT@info@nl   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@info@n1{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@info@n1{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@info@n1{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@info@n1{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@tooold=\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 xe\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
  \MT@pr@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 `ltxcntrl.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.

```

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/luatflload`.

```

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 luatflload
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@n{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@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\rightmarginkern##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  $X_{\text{E}}\text{TeX}$  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 <LaTeX-def>\MT@requires@LaTeX3
986 <pdfTeX-def>|<LaTeX-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}%
```

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.

```
993 \MT@protrusion
994 <pdfTeX-def>|<LaTeX-def> \MT@expansion
995 }
```

Interword spacing and kerning (pdfTeX 1.40).

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

Disable ligatures (pdfTeX 1.30).

```
1001 <pdfTeX-def>\MT@requires@pdfTeX5{
1002 <pdfTeX-def>|<LaTeX-def>\g@addto@macro\MT@setupfont\MT@no@ligatures
1003 <pdfTeX-def>}\relax
1004 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
1005 <debug>\MT@show@pdfannot1%
```

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

```
1006 \MT@register@font
1007 \fi
1008 }
1009 </pdfTeX-def>|<xetex-def>|<LaTeX-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.

```
1010 <*pdfTeX-def>|<LaTeX-def>
```

```

1011 \let\MT@copy@font\relax
1012 <luatex-def>\MT@requires@luatex4{\let\pdfcopyfont\copyfont}\relax
1013 <pdfTEX-def>\MT@requires@pdfTEX7{
1014 \def\MT@copy@font{%

```

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

```

1015 \xdef\MT@font@copy{\csname\MT@font/\MT@pr@context/\MT@ex@context\endcsname}%
1016 \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.

```

1017 \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
1018 \expandafter\ifx\MT@font@orig\relax
1019 \MT@exp@two@c\MT@gl@et\MT@font@orig\font@name
1020 \else
1021 \MT@exp@two@c\let\font@name\MT@font@orig
1022 \fi
1023 \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
1024 <debug>\MT@dinfo1{creating new copy: \MT@font@copy}%

```

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

```

1025 \MT@map@clist@c\MT@active@features{%
1026 \MT@exp@cs\ifx\MT@\@nameuse\MT@abbr@##1}\relax\else
1027 \def\@tempa{##1}%
1028 \MT@exp@cs\MT@map@tlist@c\MT@##1@doc@contexts}\MT@rem@from@list
1029 \fi
1030 }%
1031 \fi
1032 \MT@exp@two@c\let\MT@font\MT@font@copy

```

We only need the font identifier for letterspacing.

```

1033 \let\font@name\MT@font@copy

```

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

```

1034 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
1035 }

```

\MT@rem@from@list

```

1036 \def\MT@rem@from@list#1{%
1037 \MT@exp@cs\ifx\MT@\@tempa @#1font@list}\relax\else
1038 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
1039 \MT@font \csname \MT@\@tempa @#1font@list\endcsname
1040 \fi
1041 }
1042 <pdfTEX-def>\relax
1043 </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
\MT@encoding        letterspacing amount). With fontspec we also need to remove its internal instance
\MT@family          counter.
\MT@series 1044  (*package)
\MT@shape 1045  \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
1046    \def\MT@encoding{#1}%
\MT@size 1047    \ifMT@fontspec
1048      \edef\MT@family{\MT@scrubfeature#2()\relax}%
1049    \else
1050      \def\MT@family{#2}%
1051    \fi
1052    \def\MT@series  {#3}%
1053    \def\MT@shape   {#4}%
1054    \def\MT@size    {#5}%

\MT@familyalias    Alias family?
1055    \MT@ifdefined@n@TF{MT@\MT@family @alias}%
1056    {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
1057    {\let\MT@familyalias\@empty}%
1058  }

\MT@scrubfeature    Remove one resp. all feature counters (fontspec).
\MT@scrubfeatures  1059  \def\MT@scrubfeature#1(#2)#3\relax{#1}
1060  \def\MT@scrubfeatures#1(#2)#3\relax{%
1061    #1%
1062    \ifx\relax#3\relax\else
1063      \MT@scrubfeatures#3\relax
1064    \fi
1065  }

\ifMT@do            We check all features of the current font against the lists of the currently active
\MT@feat            font set, and set \ifMT@do accordingly.
\MT@maybe@do 1066  \newif\ifMT@do
1067  \def\MT@maybe@do#1{%
    (but only if the feature isn't globally set to false)
1068    \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).

1069    \MT@dotrue
1070    \edef@tempa{\csname MT@#1@setname\endcsname}%
1071    \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1072      \MT@ifdefined@n@TF{MT@checklist@##1}%
1073      {\csname MT@checklist@##1\endcsname}%
1074      {\MT@checklist@{##1}}}%
1075      {#1}%
1076    }%
1077    \else
1078      \MT@dofalse
1079    \fi
1080    \ifMT@do
```

\MT@feat stores the current feature.

```

1081 \def\MT@feat{#1}%
1082 \csname MT@set@#1@codes\endcsname
1083 \else
1084 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
1085 \fi
1086 }

```

\MT@dinfo@list

```

1087 <debug>\def\MT@dinfo@list#1#2#3{\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#1}: #2
1088 <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).

```

1089 \def\MT@checklist@#1#2{%
1090 <!debug> \MT@ifdefined@nT
1091 <debug> \MT@ifdefined@nTF
1092 {MT@#2list@#1@\@tempa}{%

```

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

```

1093 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
1094 \csname MT@#1\expandafter\endcsname
1095 \csname MT@#2list@#1@\@tempa\endcsname
1096 \ifMT@inlist@
1097 <debug>\MT@dinfo@list{#2}{#1}{in}%
1098 \MT@dotrue
1099 \else
1100 <debug>\MT@dinfo@list{#2}{#1}{not in}%
1101 \MT@dofalse
1102 \expandafter\MT@clist@break
1103 \fi
1104 }%

```

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.

```

1105 <debug> {\MT@dinfo@list{#2}{#1}{}}%
1106 }

```

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

```

1107 \def\MT@checklist@family#1{%
1108 <!debug> \MT@ifdefined@nT
1109 <debug> \MT@ifdefined@nTF
1110 {MT@#1list@family@\@tempa}{%
1111 \MT@exp@two@n\MT@in@clist
1112 \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
1113 \ifMT@inlist@
1114 <debug>\MT@dinfo@list{#1}{family}{in}%
1115 \MT@dotrue
1116 \else
1117 <debug>\MT@dinfo@list{#1}{family}{not in}%
1118 \MT@dofalse
1119 \ifx\MT@familyalias\empty \else
1120 \MT@exp@two@n\MT@in@clist
1121 \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
1122 \ifMT@inlist@
1123 <debug> \MT@dinfo@list{#1}{family alias}{in}%
1124 \MT@dotrue
1125 <debug>\else\MT@dinfo@list{#1}{family alias}{not in}%
1126 \fi
1127 \fi
1128 \fi
1129 \ifMT@do \else
1130 \expandafter\MT@clist@break
1131 \fi

```

```

1132 }%
1133 <debug> {\MT@info@list{#1}{family}{}}%
1134 }

```

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

```

1135 \def\MT@checklist@size#1{%
1136 <debug> \MT@ifdefined@n@T
1137 <debug> \MT@ifdefined@n@TF
1138   {MT#1list@size@\@tempa}%
1139   \MT@exp@cs\MT@in@rlist{MT#1list@size@\@tempa}%
1140   \ifMT@inlist@
1141 <debug>\MT@info@list{#1}{size}{in}%
1142   \MT@dotrue
1143   \else
1144 <debug>\MT@info@list{#1}{size}{not in}%
1145   \MT@dofalse
1146   \expandafter\MT@clist@break
1147   \fi
1148 }%
1149 <debug> {\MT@info@list{#1}{size}{}}%
1150 }

```

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

```

1151 \def\MT@checklist@font#1{%
1152 <debug> \MT@ifdefined@n@T
1153 <debug> \MT@ifdefined@n@TF
1154   {MT#1list@font@\@tempa}%

```

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

```

1155   \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
1156   \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
1157   \@tempb \csname MT#1list@font@\@tempa\endcsname
1158   \ifMT@inlist@
1159 <debug>\MT@info@list{#1}{font}{in}%
1160   \expandafter\MT@clist@break
1161   \else
1162 <debug>\MT@info@list{#1}{font}{not in}%
1163   \MT@dofalse
1164   \fi
1165 }%
1166 <debug> {\MT@info@list{#1}{font}{}}%
1167 }

```

### 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`.

```

1168 \newif\ifMT@nofamily
1169 </package>

```

`\MT@protrusion` Set up for protrusion?

```

1170 <*pdfTeX-def|xetex-def|luatex-def>
1171 \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.

```

1172 \def\MT@set@pr@codes{%
1173   \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).

```

1174 \MT@if@list@exists{%
1175 \ifMT@nofamily
1176 \MT@ifdefined@n@TF{\MT@encoding-\MT@family-settings}\relax{%
1177 \MT@info@n1{Loading generic protrusion settings for font family\MessageBreak
1178 \MT@family' (encoding: \MT@encoding).\MessageBreak
1179 For optimal results, create family-specific settings.\MessageBreak
1180 See the microtype manual for details}%
1181 \MT@gl@et@nc{\MT@encoding-\MT@family-settings}\@empty
1182 }%
1183 \fi
1184 \MT@get@font@dimen@six{%
1185 \MT@get@opt
1186 \MT@reset@pr@codes

```

Get the name of the inheritance list and parse it.

```
1187 \MT@get@inh@list
```

Set an input encoding?

```
1188 \MT@set@inputenc{c}%

```

Load additional lists?

```
1189 \MT@load@list\MT@pr@c@name
1190 \MT@set@listname

```

Load the main list.

```

1191 \MT@let@cn@tempc{MT@pr@c@\MT@pr@c@name}%
1192 \expandafter\MT@set@codes\@tempc,\relax,%
1193 }\MT@reset@pr@codes
1194 }

```

`\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`

```

1195 \def\MT@get@font@dimen@six{%
1196 \ifnum\fontdimen6\MT@font=\z@
1197 \MT@warning@n1{%
1198 Font '\MT@font' does not specify its\MessageBreak
1199 \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
1200 \@nameuse{MT@abbr\MT@feat} will not work with this font}%
1201 \expandafter@gobble
1202 \else
1203 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
1204 \expandafter@firstofone
1205 \fi
1206 }

```

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

```

1207 \def\MT@set@all@pr#1#2{%
1208 debug\MT@dinfo@n1{3}{-- lp/rp: setting all to #1/#2}%
1209 \let\MT@temp\@empty
1210 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lcode\MT@font\@tempcnta=#1}}%
1211 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rcode\MT@font\@tempcnta=#2}}%
1212 \MT@do@font\MT@temp
1213 }

```

`\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 `\microtypecontext` if necessary.

`\MT@reset@pr@codes`

```

1214 \def\MT@reset@pr@codes@\MT@set@all@pr\z@\z@
1215 \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 margin kerns. This will be activated in `\MT@set@tr@codes`.

`\MT@the@pr@code@tr`

```
1216 \def\MT@the@pr@code{\@tempcntb}

```

```

1217 <pdfTeX-def|luatex-def>
1218 <pdfTeX-def>\MT@requires@pdfTeX6
1219 <luatex-def>\MT@requires@luatex3
1220 {\def\MT@the@pr@code@tr{%
1221   \numexpr\@tempcntb+\MT@letterspace@/2\relax
1222 }
1223 }\relax
1224 </pdfTeX-def|luatex-def>

```

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

```

1225 \def\MT@set@codes#1,{%
1226   \ifx\relax#1\@empty\else
1227     \MT@split@codes #1=\relax
1228     \expandafter\MT@set@codes
1229   \fi
1230 }

```

`\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.

```

1231 \def\MT@split@codes#1=#2=#3\relax{%
1232   \def\@tempa{#1}%
1233   \ifx\@tempa\@empty \else
1234     \MT@get@slot
1235     <pdfTeX-def|luatex-def> \ifnum\MT@char > \m@ne
1236     <xetex-def> \ifx\MT@char\@empty \else
1237       \MT@get@char@unit
1238       \csname MT@\MT@feat @split@val\endcsname#2\relax
1239     \fi
1240   \fi
1241 }

```

`\MT@pr@split@val`

```

1242 \def\MT@pr@split@val#1,#2\relax{%
1243   \def\@tempb{#1}%
1244   \MT@ifempty\@tempb\relax{%
1245     \MT@scale@to@em
1246     \lcode\MT@font\MT@char=\MT@the@pr@code
1247     <debug>\MT@info@n1{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char\space: [#1]}%
1248   }%
1249   \def\@tempb{#2}%
1250   \MT@ifempty\@tempb\relax{%
1251     \MT@scale@to@em
1252     \rcode\MT@font\MT@char=\MT@the@pr@code
1253     <debug>\MT@info@n1{4}{;;; rp (\MT@char): \number\rcode\MT@font\MT@char\space: [#2]}%
1254   }%

```

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

```

1255 \MT@ifdefined@c@T\MT@pr@inh@name{%
1256   \MT@ifdefined@nT\MT@inh@\MT@pr@inh@name @\MT@char @}{%
1257     \MT@exp@cs\MT@map@tlist@c
1258     {\MT@inh@\MT@pr@inh@name @\MT@char @}%
1259     \MT@set@pr@heirs
1260   }%
1261 }%
1262 }

```

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

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

```
1263 (pdfTeX-def)\MT@requires@pdftex3{
1264 \def\MT@scale@to@em{%
1265   \@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`.

```
1266   \MT@scale\@tempcntb \@tempb \MT@dimen@six
1267   \ifnum\@tempcntb=\z@ \else
1268     \MT@scale@factor
1269   \fi
1270 }
```

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

```
1271 \def\MT@get@charwd{%
1272 (*pdfTeX-def)
1273 ^^X \MT@count=\fontcharwd\MT@font\MT@char\relax
1274 ^^Q \setbox\z@=\hbox{\MT@font \char\MT@char}%
1275 ^^Q \MT@count=\wd\z@
1276 (/pdfTeX-def)
1277 (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).

```
1278 (*xTeX-def)
1279 \ifnum\MT@char@<\z@
1280   \setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char@}%
1281   \MT@count=\wd\z@
1282 \else
1283   \MT@count=\fontcharwd\MT@font\MT@char@\relax
1284 \fi
1285 (/xTeX-def)
1286 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1287 }
```

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  $1\text{em} = \text{\fontdimen 6}$ .

```
1288 (*pdfTeX-def)
1289 \MT@requires@pdftex6{
1290   \g@addto@macro\MT@get@charwd{%
1291     \MT@ifdefined@c@T\MT@letterspace@
1292     {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1293   }
1294 }\relax
1295 }
```

No adjustment with versions 0.14f and 0.14g.

```
1296 \def\MT@scale@to@em{%
1297   \MT@count=\@tempb\relax
1298   \ifnum\MT@count=\z@ \else
1299     \MT@scale@factor
1300   \fi
1301 }
```

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

```
1302 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1303 }
1304 \pdfTeX-def
1305 \pdfTeX-def|xetex-def|luatex-def
```

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

```
1306 \*package
1307 \def\MT@get@font@dimen#1{%
1308   \ifnum\fontdimen#1\MT@font=\z@
1309     \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1310       \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1311       You should use a different 'unit' for \MT@curr@list@name}%
1312   \else
1313     \MT@count=\fontdimen#1\MT@font
1314   \fi
1315 }
```

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

```
1316 \def\MT@info@missing@char{%
1317   \MT@info@n1{Character '\the\MT@toks'
1318   ^^X   \iffontchar\MT@font\MT@char@
1319     has a width of 0pt
1320   ^^X   \else is missing\fi
1321   ^^Q   \MessageBreak (it's probably missing)
1322   \MessageBreak in font '\MT@font'. \MessageBreak
1323   Ignoring protrusion settings for this character}%
1324 }
```

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

```
1325 \def\MT@scale@factor{%
1326   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1327     \expandafter\MT@scale\expandafter \@tempcntb
1328     \csname MT@\MT@feat @factor@\endcsname \@m
1329   \fi
1330   \ifnum\@tempcntb>\csname MT@\MT@feat @max@\endcsname\relax
1331     \MT@exp@cs\MT@warn@code@too@large{\MT@\MT@feat @max}%
1332   \else
1333     \ifnum\@tempcntb<\csname MT@\MT@feat @min@\endcsname\relax
1334       \MT@exp@cs\MT@warn@code@too@large{\MT@\MT@feat @min}%
1335     \fi
1336   \fi
1337 }
```

`\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.

```
1338 \def\MT@warn@code@too@large#1{%
1339   \@tempcnta=#1\relax
1340   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1341     \expandafter\MT@scale\expandafter \@tempcnta\expandafter
1342     \@m \csname MT@\MT@feat @factor@\endcsname
1343   \fi
1344   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1345   \MT@warning@n1{The \@nameuse{\MT@abbr@\MT@feat} code \@temp\space
1346     is too large for character \MessageBreak
1347     '\the\MT@toks' in \MT@curr@list@name. \MessageBreak
1348     Setting it to the maximum of \number\@tempcnta}%
1349   \@tempcntb=#1\relax
1350 }
```

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

```
1351 \def\MT@get@opt{%
```

```

1352 \MT@set@listname

\MT@pr@factor@ Apply a factor?
\MT@sp@factor@ 1353 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 1354 \MT@let@nn{MT@\MT@feat @factor@}
1355 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1356 \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1357 \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1358 }%
1359 \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1360 }%

\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@ 1361 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1362 \MT@let@nn{MT@\MT@feat @unit@}%
1363 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1364 \MT@exp@cs@ifx{MT@\MT@feat @unit@}\@empty
1365 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1366 relative to character widths}%
1367 \else
1368 \MT@exp@cs@ifx{MT@\MT@feat @unit@}\m@ne
1369 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1370 relative to width of space}%
1371 \fi
1372 \fi
1373 }%
1374 \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
1375 }%

\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.
1376 \let\MT@get@char@unit\relax
1377 \let\MT@get@space@unit\@gobble
1378 \MT@exp@cs@ifx{MT@\MT@feat @unit@}\@empty
1379 \let\MT@get@char@unit\MT@get@charwd
1380 \else
1381 \MT@exp@cs@ifx{MT@\MT@feat @unit@}\m@ne
1382 \let\MT@get@space@unit\MT@get@font@dimen
1383 \else
1384 \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}%
1385 \fi
1386 \fi

Preset all characters? If so, we surely don't need to reset, too.
1387 \MT@ifdefined@n@T{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
1388 \csname MT@preset@\MT@feat\endcsname
1389 \MT@let@nc{MT@reset@\MT@feat @codes}\relax
1390 }%
1391 }

\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.
1392 \def\MT@get@unit#1{%
1393 \expandafter\MT@get@unit@#1 e!\@nil
1394 \ifx\x\@empty\else\let#1\x\fi
1395 \@defaultunits\@tempdima#1 pt\relax\@nnil
1396 \ifdim\@tempdima=\z@
1397 \MT@warning@n1{%
1398 Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak

```

```

1399     width. Setting factors of list `\<@nameuse{MT@MT@feat @c@name}'\MessageBreak
1400     relative to character widths instead)%
1401     \let#1\@empty
1402     \let\MT@get@char@unit\MT@get@charwd
1403     \else
1404     \MT@vinfo{... : Setting \<@nameuse{MT@abbr@MT@feat} factors relative
1405     to \the\@tempdima}%
1406     \MT@count=\@tempdima\relax
1407     \fi
1408 }
1409 \def\MT@get@unit@#1e#2#3\@nil{%
1410 \ifx\#3\\\let\x\@empty \else
1411 \if m#2%
1412 \edef\x{#1\fontdimen6\MT@font}%
1413 \else
1414 \if x#2%
1415 \edef\x{#1\fontdimen5\MT@font}%
1416 \fi
1417 \fi
1418 \fi
1419 }

```

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

```
1420 \def\MT@set@inputenc#1{%
```

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

```

1421 \def\MT@cat{#1}%
1422 \edef\@tempa{MT@MT@feat @#1@\csname MT@MT@feat @#1@name\endcsname @inputenc}%
1423 \MT@ifdefined@n@T\@tempa\MT@set@inputenc
1424 }

```

`\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.

```

1425 \MT@addto@setup{%
1426 \ifpackageloaded{inputenc}{%
1427 \ifpackageolder{inputenc}{2006/02/22}{%
1428 \def\MT@set@inputenc@{%
1429 \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1430 \MT@load@inputenc
1431 }%
1432 }{%
1433 \let\MT@set@inputenc@\MT@load@inputenc
1434 }%
1435 }%
1436 \def\MT@set@inputenc@{%
1437 \MT@warning@n1{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1438 \MessageBreak package isn't loaded. Ignoring input encoding}%
1439 }%
1440 }%
1441 }

```

`\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.

```

1442 \def\MT@load@inputenc{%
1443 \MT@cfg@catcodes
1444 (debug)\MT@edinfo@n1{1}{loading input encoding: \<@nameuse{\@tempa}}%
1445 \inputencoding{\@nameuse{\@tempa}}%
1446 }
1447 (/package)

```

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

```

1448 (*pdfTeX-def|xetex-def|luatex-def)
1449 \def\MT@set@pr@heirs#1{%
1450 \lcode\MT@font #1 =\lcode\MT@font\MT@char\relax

```

```

1451 \rptcode\MT@font #1 =\rptcode\MT@font\MT@char\relax
1452 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1453 <debug>\MT@info@n1{4}{;;; lp/rp (#1): \number\lptcode\MT@font\MT@char\space/%
1454 <debug> \number\rptcode\MT@font\MT@char\space}%
1455 }

```

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

```

\MT@preset@pr 1456 \def\MT@preset@pr{%
1457 \expandafter\expandafter\expandafter\MT@preset@pr@
1458 \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1459 }
1460 \def\MT@preset@pr@#1,#2\@nil{%
1461 \ifx\MT@pr@unit@\@empty
1462 \MT@warn@preset@tewidth{pr}%
1463 \let\MT@preset@aux\MT@preset@aux@factor
1464 \else
1465 \def\MT@preset@aux{\MT@preset@aux@space2}%
1466 \fi
1467 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1468 \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1469 \MT@set@all@pr\@tempa\@tempb
1470 }

```

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

```

\MT@preset@aux@factor 1471 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 1472 \@tempcntb=#1\relax
1473 \MT@scale@factor
1474 \edef#2{\number\@tempcntb}%
1475 }
1476 \def\MT@preset@aux@space#1#2#3{%
1477 \def\@tempb{#2}%
1478 \MT@get@space@unit#1%
1479 \MT@scale@to@em
1480 \edef#3{\number\@tempcntb}%
1481 }

```

`\MT@warn@preset@tewidth`

```

1482 \def\MT@warn@preset@tewidth#1{%
1483 \MT@warning@n1{%
1484 Cannot preset characters relative to their widths\MessageBreak
1485 for \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1486 \MessageBreak relative to lem instead}%
1487 }
1488 </pdfTeX-def|xetex-def|luatex-def>

```

### 14.2.2 Expansion

`\MT@expansion` Set up for expansion?

```

1489 <*pdfTeX-def|luatex-def>
1490 \def\MT@expansion{\MT@maybe@do{ex}}

```

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

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

```

1491 \def\MT@set@ex@codes@s{%
1492 \MT@if@list@exists{%
1493 \MT@get@ex@opt
1494 \let\MT@get@char@unit\relax
1495 \MT@reset@ef@codes
1496 \MT@get@inh@list
1497 \MT@set@inputenc{c}%

```

```

1498 \MT@load@list\MT@ex@cc@name
1499 \MT@set@listname
1500 \MT@let@cn\@tempc{MT@ex@cc@MT@ex@cc@name}%
1501 \expandafter\MT@set@codes\@tempc,\relax,%
1502 \MT@expandfont
1503 }\relax
1504 }
1505 </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.
1506 <package>\newif\ifMT@nonselected
1507 <*pdfTeX-def|luatex-def>
1508 \def\MT@set@ex@codes@n{%
1509 \MT@nonselectedtrue
1510 \MT@if@list@exists
1511 \MT@get@ex@opt
1512 {%
1513 \let\MT@stretch@ \MT@stretch
1514 \let\MT@shrink@ \MT@shrink
1515 \let\MT@step@ \MT@step
1516 \let\MT@auto@ \MT@auto
1517 \let\MT@ex@factor@ \MT@ex@factor
1518 }%
1519 \MT@reset@ef@codes
1520 \MT@expandfont
1521 \MT@nonselectedfalse
1522 }
\MT@set@ex@codes Default is non-selected. It can be changed in the package options.
1523 \let\MT@set@ex@codes\MT@set@ex@codes@n
\MT@expandfont Expand the font.
1524 <luatex-def>\MT@requires@luatex4{\let\pdfFontExpand\expandGlyphsInFont}\relax
1525 \def\MT@expandfont{%
1526 \pdfFontExpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1527 }
\MT@set@all@ex At first, all expansion factors for the characters will be set to 1000 (respectively the
\MT@reset@ef@codes factor of this font).
1528 \def\MT@set@all@ex#1{%
1529 <debug>\MT@diinfo@n1{3}{-- ex: setting all to \number#1}%
1530 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1531 }
1532 \def\MT@reset@ef@codes@{\MT@set@all@ex\MT@ex@factor@}
\MT@reset@ef@codes However, this is only necessary for pdfTeX versions prior to 1.20, or LuaTeX < 0.90
                    (actually, I think, 0.87).
1533 <pdfTeX-def>\MT@requires@pdfTeX4
1534 <luatex-def>\MT@requires@luatex5
1535 {
1536 \def\MT@reset@ef@codes{%
1537 \ifnum\MT@ex@factor@=\@m \else
1538 \MT@reset@ef@codes@
1539 \fi
1540 }
1541 }{
1542 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1543 }

```

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

```
1544 \def\MT@ex@split@val#1\relax{%
1545   \@tempcntb=#1\relax
```

Take an optional factor into account.

```
1546   \ifnum\MT@ex@factor@=\@m \else
1547     \MT@scale\@tempcntb \MT@ex@factor@ \@m
1548   \fi
1549   \ifnum\@tempcntb > \MT@ex@max
1550     \MT@warn@ex@too@large\MT@ex@max
1551   \else
1552     \ifnum\@tempcntb < \MT@ex@min
1553       \MT@warn@ex@too@large\MT@ex@min
1554     \fi
1555   \fi
1556   \efcode\MT@font\MT@char=\@tempcntb
1557 <debug>\MT@dinfo@n1{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%
```

Heirs, heirs, I love thy heirs.

```
1558   \MT@ifdefined@c@T\MT@ex@inh@name{%
1559     \MT@ifdefined@nT{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1560       \MT@exp@cs\MT@map@tlist@c{\MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
1561     }%
1562   }%
1563 }
```

\MT@warn@ex@too@large

```
1564 \def\MT@warn@ex@too@large#1{%
1565   \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1566     character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1567     Setting it to the maximum of \number#1}%
1568   \@tempcntb=#1\relax
1569 }
```

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

```
\MT@ex@factor@ 1570 \def\MT@get@ex@opt{%
\MT@stretch@ 1571   \MT@set@listname
1572   \MT@ifdefined@n@TF{\MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1573     \MT@let@cn\MT@ex@factor@\MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1574     \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 1575   }%
1576     \let\MT@ex@factor@\MT@ex@factor
1577   }%
1578   \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1579   \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1580   \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
1581   \def\@tempa{autoexpand}%
1582   \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1583   \MT@ifdefined@n@T{\MT@ex@c@\MT@ex@c@name @preset}{%
1584     \MT@preset@ex
1585     \let\MT@reset@ef@codes\relax
1586   }%
1587 }
```

\MT@get@ex@opt@

```
1588 \def\MT@get@ex@opt@#1#2{%
1589   \MT@ifdefined@n@TF{\MT@ex@c@\MT@ex@c@name @#1}{%
1590     \MT@let@nn{\MT@#1@}{\MT@ex@c@\MT@ex@c@name @#1}%
1591     \MT@vinfo{... : #2}%
1592   }%
1593   \MT@let@nn{\MT@#1@}{\MT@#1}%
1594   }%
1595 }
```

`\MT@set@ex@heirs`

```
1596 \def\MT@set@ex@heirs#1{%
1597   \efcode\MT@font#1=\efcode\MT@font\MT@char
1598 <debug>\MT@dinfoln{2}{-- heir of \MT@char: #1}%
1599 <debug>\MT@dinfoln{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1600 }
```

`\MT@preset@ex`

```
1601 \def\MT@preset@ex{%
1602   \@tempcntb=\csname MT@ex@cc\MT@ex@cc@name @preset\endcsname\relax
1603   \MT@scale@factor
1604   \MT@set@all@ex\@tempcntb
1605 }
1606 </pdfTeX-def|LaTeX-def>
```

### 14.2.3 Interword spacing (glue)

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

```
1607 <*pdfTeX-def>
1608 \MT@requires@pdfTeX6{
1609 \def\MT@spacing{\MT@maybe@do{sp}}
```

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

```
1610 \def\MT@set@sp@codes{%
1611   \MT@if@list@exists{%
1612     \MT@get@font@dimen@six{%
1613       \MT@get@opt
1614       \MT@reset@sp@codes
1615       \MT@get@inh@list
1616       \MT@set@inputenc{c}%
1617       \MT@load@list\MT@sp@cc@name
1618       \MT@set@listname
1619       \MT@let@cn\@tempc{MT@sp@cc\MT@sp@cc@name}%
1620       \expandafter\MT@set@codes\@tempc,\relax,}%
1621   }\MT@reset@sp@codes
1622 }
```

`\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).

```
1623 \def\MT@sp@split@val#1,#2,#3\relax{%
1624   \def\@tempb{#1}%
1625   \MT@ifempty\@tempb\relax{%
1626     \MT@get@space@unit2%
1627     \MT@scale@to@em
1628     \knbscode\MT@font\MT@char=\@tempcntb
1629 <debug>\MT@dinfoln{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1630   }%
1631   \def\@tempb{#2}%
1632   \MT@ifempty\@tempb\relax{%
1633     \MT@get@space@unit3%
1634     \MT@scale@to@em
1635     \stbscode\MT@font\MT@char=\@tempcntb
1636 <debug>\MT@dinfoln{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1637   }%
1638   \def\@tempb{#3}%
1639   \MT@ifempty\@tempb\relax{%
1640     \MT@get@space@unit4%
1641     \MT@scale@to@em
1642     \shbscode\MT@font\MT@char=\@tempcntb
1643 <debug>\MT@dinfoln{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1644   }%
1645   \MT@ifdefined@cT\MT@sp@inh@name{%
1646     \MT@ifdefined@nT{MT@inh\MT@sp@inh@name @\MT@char @}{%
```

```

1647 \MT@exp@cs\MT@map@tlist@c{MT@inh@MT@sp@inh@name @\MT@char @}\MT@set@sp@heirs
1648 }%
1649 }%
1650 }

```

\MT@set@sp@heirs

```

1651 \def\MT@set@sp@heirs#1{%
1652 \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1653 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1654 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1655 <debug>\MT@dinfoln{2}{-- heir of \MT@char: #1}%
1656 <debug>\MT@dinfoln{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1657 <debug> \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1658 }

```

\MT@set@all@sp

```

\MT@reset@sp@codes 1659 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1660 <debug>\MT@dinfoln{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1661 \let\MT@temp\@empty
1662 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
1663 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1664 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1665 \MT@do@font\MT@temp
1666 }
1667 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@z@z@z@}
1668 \let\MT@reset@sp@codes\relax

```

\MT@preset@sp

```

\MT@preset@sp@ 1669 \def\MT@preset@sp{%
1670 \expandafter\expandafter\expandafter\MT@preset@sp@
1671 \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1672 }
1673 \def\MT@preset@sp@#1.#2.#3\@nil{%
1674 \ifx\MT@sp@unit@\@empty
1675 \MT@warn@preset@twidth{sp}%
1676 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
1677 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
1678 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
1679 \else
1680 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
1681 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
1682 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
1683 \fi
1684 \MT@set@all@sp\@tempa\@tempc\@tempb
1685 }
1686 }\relax

```

#### 14.2.4 Additional kerning

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

```

1687 \MT@requires@pdftex6{
1688 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

1689 \def\MT@set@kn@codes{%
1690 \MT@if@list@exists{%
1691 \MT@get@font@dimen@six{%
1692 \MT@get@opt
1693 \MT@reset@kn@codes
1694 \MT@get@inh@list
1695 \MT@set@inputenc{c}%
1696 \MT@load@list\MT@kn@c@name
1697 \MT@set@listname

```

```

1698     \MT@let@cn\@tempc{MT@kn@c@MT@kn@c@name}%
1699     \expandafter\MT@set@codes\@tempc,\relax,%
1700   }\MT@reset@kn@codes
1701 }

```

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

```

1702 \def\MT@kn@split@val#1,#2\relax{%
1703   \def\@tempb{#1}%
1704   \MT@ifempty\@tempb\relax{%
1705     \MT@get@space@unit2%
1706     \MT@scale@to@em
1707     \knbcode\MT@font\MT@char=\@tempcntb
1708   }%
1709   \def\@tempb{#2}%
1710   \MT@ifempty\@tempb\relax{%
1711     \MT@get@space@unit2%
1712     \MT@scale@to@em
1713     \knacode\MT@font\MT@char=\@tempcntb
1714   }%
1715   \MT@exp@cs\MT@map@tlist@{MT@inh@MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1716 }%
1717 \MT@ifdefined@c@T\MT@kn@inh@name{%
1718   \MT@ifdefined@n@T{MT@inh@MT@kn@inh@name @\MT@char @}%
1719   \MT@exp@cs\MT@map@tlist@{MT@inh@MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1720 }%
1721 }%
1722 }

```

\MT@set@kn@heirs

```

1723 \def\MT@set@kn@heirs#1{%
1724   \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1725   \knacode\MT@font#1=\knacode\MT@font\MT@char
1726 }%
1727 \MT@exp@cs\MT@map@tlist@{MT@inh@MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1728 }%
1729 }

```

\MT@set@all@kn

```

\MT@reset@kn@codes 1730 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1731 \MT@exp@cs\MT@map@tlist@{MT@inh@MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1732 }%
1733 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbcode\MT@font\@tempcnta=#1\relax}}%
1734 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knacode\MT@font\@tempcnta=#2\relax}}%
1735 \MT@do@font\MT@temp
1736 }
1737 \def\MT@reset@kn@codes@\MT@set@all@kn\z@\z@}
1738 \let\MT@reset@kn@codes\relax

```

\MT@preset@kn

```

\MT@preset@kn@ 1739 \def\MT@preset@kn{%
1740   \expandafter\expandafter\expandafter\MT@preset@kn@
1741   \csname MT@kn@c@MT@kn@c@name @preset\endcsname\@nil
1742 }%
1743 \def\MT@preset@kn@#1,#2\@nil{%
1744   \ifx\MT@kn@unit@\empty
1745     \MT@warn@preset@twidth{kn}%
1746     \let\MT@preset@aux\MT@preset@aux@factor
1747   \else
1748     \def\MT@preset@aux{\MT@preset@aux@space2}%
1749   \fi
1750   \MT@ifempty{#1}\let\@tempa\@empty{\MT@preset@aux{#1}\@tempa}%
1751   \MT@ifempty{#2}\let\@tempb\@empty{\MT@preset@aux{#2}\@tempb}%
1752   \MT@set@all@kn\@tempa\@tempb
1753 }%
1754 }%

```

1755 *(/pdf<sub>tex</sub>-def)*

### 14.2.5 Tracking

This only works with pdf<sub>TEX</sub> 1.40 or Lua<sub>TEX</sub> 0.62.

```
1756 (*pdftex-def|luatex-def)
1757 (pdftex-def)\MT@requires@pdftex6
1758 (luatex-def)\MT@requires@luatex3
1759 {
```

`\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@tr@font@list 1760 \let\MT@tr@font@list\@empty
1761 \def\MT@tracking@{%
1762   \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
1763   \ifMT@inlist@else
1764     \MT@maybe@do{tr}%
1765     \ifMT@do@else
1766       \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1767     \fi
1768   \fi
1769 }
1770 (/pdftex-def|luatex-def)
1771 (pdftex-def|luatex-def|letterspace)\let\MT@tracking
1772 (pdftex-def|luatex-def) \MT@tracking@
1773 (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.

```
1774 (*pdftex-def|luatex-def|letterspace)
1775 \def\MT@set@tr@codes{%
1776 (*pdftex-def|luatex-def)
1777   \MT@vinfo{Tracking font '\MT@font'\on@line}%
1778   \MT@get@font@dimen@six{%
1779     \MT@if@list@exists
1780     \MT@get@tr@opt
1781     \relax
1782 (/pdftex-def|luatex-def)
1783     \MT@ifdefined@c@TF\MT@letterspace@relax{\let\MT@letterspace@ \MT@letterspace}%
1784     \ifnum\MT@letterspace@=\z@
```

Zero tracking requires special treatment.

```
1785   \MT@set@tr@zero
1786   \else
1787 (pdftex-def|luatex-def)   \MT@vinfo{... Tracking by \number\MT@letterspace@}%
```

Letterspacing only works in PDF mode.

```
1788   \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.

```
1789   \xdef\MT@lsfont{\csname\expandafter\string\font@name
1790     /\number\MT@letterspace@ ls\endcsname}%
1791   \expandafter\ifx\MT@lsfont\relax
1792 (debug)\MT@dinfo@nl{1}{... new letterspacing instance}%
```

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

```
1793   \MT@get@ls@basefont
```

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

```

1794 (*luatex-def|letterspace)
1795 \MT@if@fontspec@font{%
1796 <luatex-def&debug>\MT@info@n1{1}{... fontspec font: \MessageBreak
1797 <luatex-def&debug> \expandafter\fontname\font@name}%
1798 \ifnum\MT@letterspace@<z@\def\MT@minus{-}\else\let\MT@minus\empty\fi
1799 \global\expandafter\font\MT@font=%
1800 \expandafter\MT@exp@two@c\expandafter\MT@font@fontspec@font
1801 \expandafter\fontname\expandafter\font@name\space \@nil
1802 }{%
1803 </luatex-def|letterspace>
1804 <luatex-def&debug>\MT@info@n1{1}{... legacy font}%
1805 \global\expandafter\letterspacefont\MT@font\font@name\MT@letterspace@
1806 <luatex-def|letterspace> }%

```

Scale interword spacing (not configurable in letterspace).

```

1807 (*pdfTeX-def|luatex-def)
1808 \MT@ifdefined@c@TF\MT@tr@ispace
1809 {\let\@tempa\MT@tr@ispace}%
1810 {\edef\@tempa{\MT@letterspace@*,,}}%
1811 \MT@ifdefined@c@TF\MT@tr@ospace
1812 {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1813 {\edef\@tempa{\@tempa,,,}}%
1814 \expandafter\MT@tr@set@space\@tempa,%
1815 </pdfTeX-def|luatex-def>
1816 (*letterspace)
1817 % spacing = <letterspace amount>*,,
1818 \fontdimen2\MT@font=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1819 * \fontdimen2\MT@font/1000\relax
1820 </letterspace>

```

Adjust outer kerning (microtype only).

```

1821 (*pdfTeX-def|luatex-def)
1822 \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
1823 \expandafter\MT@tr@set@okern\@tempa,%

```

Disable ligatures (not configurable in letterspace).

```

1824 \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1825 </pdfTeX-def|luatex-def>
1826 (*letterspace)
1827 % no ligatures = {f}
1828 \tagcode\MT@font`f=\m@ne
1829 </letterspace>

```

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

```

1830 <luatex-def|letterspace> \MT@if@fontspec@font\relax{%
1831 <debug>\MT@info@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%
1832 \MT@do@font{\lcode\MT@font\@tempcnta=\numexpr\MT@letterspace@/2\relax
1833 \rcode\MT@font\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
1834 \let\MT@the@pr@code\MT@the@pr@code@tr
1835 <luatex-def|letterspace> }%
1836 \fi

```

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

```

1837 \aftergroup\MT@set@font
1838 <pdfTeX-def|luatex-def> \let\MT@font\MT@font
1839 <luatex-def> \MT@if@fontspec@font\MT@font\relax

```

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

```

\MT@curr@ls 1840 \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
1841 \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.

```
1842 (*pdfTeX-def|LaTeX-def)
1843   \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
1844   \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1845   \MT@tr@outer@l
1846 (/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'.

```
1847   \ifx\MT@ls@adjust@empty
1848 (!letterspace)   % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}
1849   \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1850   \MT@ls@outer@k
```

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

```
1851 (*pdfTeX-def|LaTeX-def)
1852   \else
1853     \MT@outer@kern=\expandafter\expandafter\expandafter@firstoftwo
1854     \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1855     \ifdim\MT@outer@kern=z@ \else \MT@ls@outer@k \fi
1856     \MT@outer@kern=\expandafter\expandafter\expandafter@secondoftwo
1857     \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1858 (/pdfTeX-def|LaTeX-def)
1859 (*letterspace)
1860   \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1861   \MT@afteraftergroup{%
1862     \MT@set@curr@ok
1863     \noexpand\MT@ls@outer@k
1864   }%
1865 (/letterspace)
1866   \fi
1867 (*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).

```
1868   \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.

```
1869   \MT@afteraftergroup{%
1870     \MT@set@curr@os
1871     \MT@set@curr@ok
1872     \noexpand\MT@tr@outer@r
1873   }%
1874 (/pdfTeX-def|LaTeX-def)
1875   \fi
1876 (pdfTeX-def|LaTeX-def) }%
1877 }
```

`\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.)

```
1878 \def\MT@afteraftergroup#1{%
1879 (!letterspace) \MT@maybe@gobble@with@tikz{%
1880   \MT@ifdefined@n@TF{MT@aftergroup@ \number\currentgrouplevel}\relax{%
1881     \MT@exp@cs\xdef{MT@aftergroup@ \number\currentgrouplevel}%
1882     {\MT@exp@cs\MT@gl@et{MT@aftergroup@ \number\currentgrouplevel}\noexpand\undefined#1}%
1883     \expandafter\aftergroup\expandafter\aftergroup\MT@exp@cs\aftergroup
1884     {MT@aftergroup@ \number\currentgrouplevel}%
```

```

1885     }%
1886     !letterspace }%
1887 }
1888 </pdfTeX-def|luatex-def|letterspace>

\MT@!s@fontspec@colon    Add the kernfactor feature to a font loaded by fontspec (we might have to add
\MT@!s@fontspec@font    the colon ourselves).
1889 <*luatex-def|letterspace>
1890 \def\MT@!s@fontspec@colon#1:#2:#3:#4@nil{\ifx\#3\#1:#2\else#1:#2:#3\fi}
1891 \def\MT@!s@fontspec@font#1 #2@nil{%
1892   "\MT@!s@fontspec@colon#1:::\relax@nil
1893   kernfactor=\MT@minus \ifnum\MT@!letterspace@=1000 1\else 0.%
1894     \ifnum\MT@minus\MT@!letterspace@<100 0\fi
1895     \ifnum\MT@minus\MT@!letterspace@<10 0\fi
1896     \number\MT@minus\MT@!letterspace@ \fi;"
1897   \ifx\#2\ at \f@size pt\else#2\fi\relax
1898 }
1899 </luatex-def|letterspace>

\MT@get@tr@opt    Various settings (only for the microtype version).
1900 <*pdfTeX-def|luatex-def>
1901 \def\MT@get@tr@opt{%
1902   \MT@set@!istname
1903   \MT@i fdefined@n@T{MT@tr@c@\MT@tr@c@name}{%
1904     \MT@!et@cn\MT@!letterspace{MT@tr@c@\MT@tr@c@name}%

\MT@tr@unit@    Different unit?
1905   \MT@i fdefined@n@T{MT@tr@c@\MT@tr@c@name @unit}{%
1906     \MT@!et@cn\MT@tr@unit@{MT@tr@c@\MT@tr@c@name @unit}%
1907     \ifdim\MT@tr@unit@=1em
1908       \let\MT@tr@unit@\undefined
1909     \else
1910       \MT@!et@cn\@tempb{MT@tr@c@\MT@tr@c@name}%
1911       \MT@get@unit\MT@tr@unit@
1912       \let\MT@tr@factor@\@m
1913       \MT@scale@to@em
1914       \edef\MT@!letterspace{\number\@tempcntb}%
1915       \fi
1916     }%
1917   }%

\MT@tr@ispace    Adjust interword spacing.
\MT@tr@ospace 1918 \MT@get@tr@opt@{spacing} {ispace}%
1919 \MT@get@tr@opt@{outerspacing}{ospace}%

\MT@tr@okern    Adjust outer kerning.
1920 \MT@get@tr@opt@{outerkerning}{okern}%

\MT@tr@ligatures    Which ligatures should we disable (empty means all, undefined none)?
1921 \MT@get@tr@opt@{noligatures} {ligatures}%
1922 }

\MT@get@tr@opt@
1923 \def\MT@get@tr@opt@#1#2{%
1924   \MT@i fdefined@n@T{MT@tr@c@\MT@tr@c@name @#1}{%
1925     { \MT@!et@nn{MT@tr@#2}{MT@tr@c@\MT@tr@c@name @#1}}%
1926   }
1927 </pdfTeX-def|luatex-def>

\MT@set@!sfont    Redefine \font@name, which will be called a second later (in \selectfont).
1928 <*pdfTeX-def|luatex-def|letterspace>
1929 <plain>\MT@requires@!atex2{
1930 \def\MT@set@!sfont{\MT@exp@two@c\let\font@name\MT@!sfont}

```

`\lssstyle` Disable the tests whether the font should be letterspaced, then trigger the setup. Only `\textls` can be used in math mode (`\lssstyle` may be used inside another text switch, of course). Still, we have to ensure that math fonts are set up again. Setting `\glb@currsiz` to `\@empty` (our previous solution) could throw us into an infinite loop (e.g., with the `psnfss` packages, via `\every@math@size`), so we issue `\glb@settings` instead.

```
1931 \DeclareRobustCommand\lssstyle{%
1932   \not@math@alphabet\lssstyle\textls
1933   (pdfTeX-def|LaTeX-def) \MT@maybe@gobble@with@tikz{\aftergroup\glb@settings}%
1934   (pdfTeX-def|LaTeX-def) \def\MT@feat{tr}%
1935   \let\MT@tracking\MT@set@tr@codes
1936   \selectfont
1937 }
```

Now the definitions for the letterspace package with plain T<sub>E</sub>X.

```
1938 (*plain)
1939 {}{
1940 \def\MT@set@lsfont{\MT@lsfont}
1941 \def\lssstyle{%
1942   \begingroup
1943   \escapechar@m@ne
1944   \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1945   \MT@set@tr@codes
1946   \endgroup
1947 }
1948 \let\textls\@undefined
1949 \let\lslig\@undefined
1950 }
1951 (/plain)
```

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

```
1952 \DeclareRobustCommand\lslig[1]{%
1953   {\MT@ifdefined@c@TF\MT@curr@ls{%
1954     \escapechar@m@ne
1955     \MT@get@ls@basefont
1956     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1957     \kern\MT@outer@kern
1958     \font@name #1%
1959     \kern\MT@outer@kern
1960   }{#1}}%
1961 }
```

`\MT@ls@basefont` pdfT<sub>E</sub>X cannot letterspace fonts that already are letterspaced. Therefore, we have to save the base font in `\font name`@base.

The previous solution (checking the macro's meaning with `\pdfmatch`), where we were loading the base font via the `\font` primitive again, would destroy all previously set up micro-typographic features of the font.

```
1962 \def\MT@get@ls@basefont{%
1963   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1964   \expandafter\ifx\MT@ls@basefont\relax
1965     \MT@exp@two@c\MT@gl@et\MT@ls@basefont\font@name
1966   \else
1967     (debug)\MT@dinfo@n1{1}{... fixing base font}%
1968     \MT@exp@two@c\let\font@name\MT@ls@basefont
1969     \fi
1970 }
```

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

`\MT@set@tr@zero`

```
1971 \def\MT@set@lsbasefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
```

```

1972 \def\MT@set@tr@zero{%
1973 <debug>\MT@info@n1{1}{... zero tracking}%
1974 \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1975 \expandafter\ifx\MT@ls@basefont\relax \else
1976 <debug>\MT@info@n1{1}{... fixing base font}%
1977 \aftergroup\MT@set@ls@basefont
1978 \fi
1979 }
1980 </pdfTeX-def|LaTeX-def|letterspace>

```

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

```

1981 <*pdfTeX-def|LaTeX-def>
1982 <pdfTeX-def>\MT@requires@pdfTeX7{
1983 \def\MT@tr@noligatures{%
1984 \ifx\MT@tr@ligatures\@empty
1985 \MT@noligatures@\MT@lsfont\@undefined
1986 \else
1987 \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1988 \fi
1989 }
1990 <*pdfTeX-def>
1991 }{
1992 \def\MT@tr@noligatures{%
1993 \MT@warning@n1{%
1994 Disabling selected ligatures is only possible since\MessageBreak
1995 pdfTeX 1.40.4. Disabling all ligatures instead}%
1996 \MT@glet\MT@tr@noligatures\relax
1997 }
1998 }
1999 </pdfTeX-def>

```

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

```
2000 \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.

```

2001 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
2002 <debug>\MT@info@n12{... orig. space: \the\fontdimen2\MT@lsfont,
2003 <debug> \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
2004 <debug> \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
2005 \let\MT@temp\@empty
2006 \MT@tr@set@space@{#1}{#4}{2}\@empty
2007 \MT@tr@set@space@{#2}{#5}{3}\@pplus
2008 \MT@tr@set@space@{#3}{#6}{4}\@minus
2009 \MT@glet@c{MT@outer@space\expandafter\string\font@name}\MT@temp
2010 <debug>\MT@info@n12{... inner space: \the\fontdimen2\MT@lsfont,
2011 <debug> \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
2012 <debug>\MT@info@n12{... outer space: \MT@temp}%
2013 }

```

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

```

2014 \def\MT@tr@set@space@#1#2#3#4{%
2015 \MT@ifempty{#2}{%
2016 \MT@ifempty{#1}{%
2017 \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
2018 }{%
2019 \MT@tr@set@space@@{#1}{#3}{1000}%
2020 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
2021 \fontdimen#3\MT@lsfont=\@tempdima
2022 }%
2023 }{%
2024 \MT@tr@set@space@@{#2}{#3}{2000}%

```

```

2025 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
2026 \MT@ifempty{#1}\relax{%
2027   \MT@tr@set@space@{#1}{#3}{1000}%
2028   \fontdimen#3\MT@lsfont=\@tempdima
2029 }%
2030 }%
2031 }

```

`\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.

```

2032 \def\MT@tr@set@space@#1#2#3{%
2033   \MT@test@ast#1*\@nil{%
2034     \MT@ifdefined@c@TF\MT@tr@unit@
2035     {\edef\@tempb{#1}\MT@scale@to@em}
2036     {\@tempcntb=#1\relax}%
2037     \@tempdima=\dimexpr \dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
2038     -\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).

```

2039   \ifnum#2=\tw@
2040     \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
2041   \fi
2042   \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
2043 }{%
2044   \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
2045   \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
2046 }%
2047 (debug)\MT@dinfo@n13{... : font dimen #2 (#1): \the\@tempdima}%
2048 }

```

`\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).

```

2049 \def\MT@tr@outer@1{%
2050   \ifhmode
2051     \ifdim\lastskip>5sp
2052       \edef\x{\the\lastskip minus 0pt}%
2053       \setbox\z@\hbox{\MT@outer@space=\x}%
2054       \ifdim\wd\z@>\z@
2055         (debug)\MT@dinfo2{[[[ adjusting pre space: \the\MT@outer@space}%
2056         \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

2057   \let\MT@ls@outer@k\relax
2058   \else

```

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

```

2059   \ifdim\lastskip=%
2060     \ifnum\spacefactor<2000
2061       \spaceskip
2062     \else
2063       \ifdim\xspaceskip=\z@
2064         \dimexpr\spaceskip+\fontdimen7\font@name\relax
2065       \else
2066         \xspaceskip
2067       \fi
2068     \fi
2069 (debug)\MT@dinfo2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
2070     \unskip \hskip\MT@outer@space\relax
2071     \let\MT@ls@outer@k\relax
2072   \fi
2073   \fi
2074   \fi
2075   \fi

```

```

2076 }
\MT@tr@outer@next    microtype also adjusts spacing. The following is borrowed from soul. I've added the
\MT@tr@outer@r      cases for italic correction, since tracking may also be triggered by text commands
                    (e.g., \textsc).
2077 \def\MT@tr@outer@r{%
2078   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
2079 }
\MT@if@outer@next    We avoid using \ifx tests, in case \MT@tr@outer@next is \let to \fi etc.
2080 \def\MT@if@outer@next#1{%
2081   \ifx\MT@tr@outer@next#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
2082 }
\MT@tr@outer@r@
2083 \def\MT@tr@outer@r@{%
2084   \def\MT@temp*{%
                Don't adjust in math mode. There was a tricky bug when \textls was the last
                command in a \mathchoice group.
2085   \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).
2086     \ifnum\currentgrouptype=10 \else
2087       \def\MT@temp*##1{\ifhmode\hskip\MT@outer@space
2088 (debug)\MT@dinfor2{}} adjusting post space (1): \the\MT@outer@space}%
2089       \fi}%
2090       \expandafter\ifcat\expandafter\noexpand\csname MT@tr@outer@next\endcsname\egroup

2091       \ifhmode\unkern\fi\egroup
2092       \MT@set@curr@ok \MT@set@curr@os
2093       \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=}
2094       \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.
2095       \MT@if@outer@next\maybe@ic{%
2096         \MT@set@curr@ok \MT@set@curr@os
2097         \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=}
2098       }{%

                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).
2099       \MT@if@outer@next\check@icr{%
2100         \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\MT@temp=}
2101       }{%
2102         \MT@if@outer@next\@sptoken{%
2103           \def\MT@temp* {\ifhmode\hskip\MT@outer@space
2104 (debug)\MT@dinfor2{}} adjusting post space (2): \the\MT@outer@space}%
2105           \fi}%
2106         }{%
2107         \MT@if@outer@next~{%
2108           \def\MT@temp*~{\nobreak\hskip\MT@outer@space
2109 (debug)\MT@dinfor2{}} adjusting post space (3): \the\MT@outer@space}%
2110         }
2111       }
2112       \MT@if@outer@next\ \relax{%
2113       \MT@if@outer@next\space\relax{%

```

```
2114 \MT@if@outer@next\xobeysp\relax{%
```

xspace requires special treatment.

```
2115 \MT@if@outer@next\xspace{%
2116 \def\MT@temp*\xspace{\MT@xspace}%
2117 }%
```

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

```
2118 \def\MT@temp*{\ifdim\MT@outer@kern=z@\else\MT@ls@outer@k
2119 <debug>\MT@dinfor2{--- adjusting post kern: \the\MT@outer@kern}%
2120 \fi}%
2121 \MT@let@nc{\MT@tr@outer@next}\relax
2122 }}}}]]}}\fi
2123 \fi\fi
2124 \MT@temp*%
2125 }
```

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

```
\MT@tr@outer@icr@ 2126 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
2127 \def\MT@tr@outer@icr@{%
2128 \let\@let@token= \MT@tr@outer@next
2129 \maybe@ic@
2130 }
```

`\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.

```
2131 \def\MT@xspace{\futurelet\@let@token\MT@xspace@}
2132 \def\MT@xspace@{\@xspace@firsttrue\xspace
2133 \ifdim\lastskip>5sp
2134 \unskip \hskip\MT@outer@space
2135 \else
2136 \ifdim\MT@outer@kern=z@\else\MT@ls@outer@k \fi
2137 \fi
2138 }
```

For older pdfTeX versions and LuaTeX, throw an error.

```
2139 }{
2140 \DeclareRobustCommand\lsstyle{%
2141 \MT@error{Letterspacing only works with \MT@engine tex version
2142 <pdftex-def> 1.40%
2143 <luatex-def> 0.62%
2144 \MessageBreak or newer}
2145 {Upgrade \MT@engine tex, or try the `soul' package instead.}%
2146 \MT@glet\lsstyle\relax
2147 }
2148 }
```

And for XeTeX, too.

```
2149 </pdftex-def|luatex-def>
2150 <*xetex-def>
2151 \DeclareRobustCommand\lsstyle{%
2152 \MT@error{Letterspacing currently doesn't work with xetex}
2153 {Run pdftex or luatex, or use the `soul' package instead.}%
2154 \MT@glet\lsstyle\relax
2155 }
2156 </xetex-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.

```
2157 <*package|letterspace>
2158 \DeclareRobustCommand\textls{%
2159 \ifstar{\let\MT@ls@adjust@\MT@ls@adjust@empty\MT@textls}%

```

```

2160         {\let\MT@ls@adjust@MT@ls@adjust@relax\MT@textls}%
2161 }

\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.

2162 \newcommand\MT@textls[2][ ]{%
2163   \ifmmode
2164     \nfss@text{\MT@ls@set@ls{#1}\lssstyle#2}%
2165   \else
2166     \hmode@bgroup
2167       \MT@ls@set@ls{#1}%
2168       \lssstyle #2%
2169       \expandafter
2170     \egroup
2171   \fi
2172 }

\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 2173 \def\MT@ls@adjust@empty{\let\MT@ls@adjust@empty}
\MT@ls@set@ls    2174 \def\MT@ls@adjust@relax{\let\MT@ls@adjust@relax}
2175 \def\MT@ls@set@ls#1{%
2176   \MT@ifempty{#1}%
2177   {\let\MT@letterspace@undefined}%
2178   {\KV@sp@def\MT@letterspace@{#1}%
2179     \edef\MT@letterspace@{\number\MT@letterspace@}%
2180     \MT@ls@too@large\MT@letterspace@}%
2181   \MT@ls@adjust@
2182 }

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

2183 \def\MT@ls@too@large#1{%
2184   \ifnum#1>\MT@tr@max
2185     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
2186     \let#1\MT@tr@max
2187   \else
2188     \ifnum#1<\MT@tr@min
2189       \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
2190       \let#1\MT@tr@min
2191     \fi
2192   \fi
2193 }

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

2194 \newdimen\MT@outer@kern
2195 /package| letterspace
2196 *pdftex-def| luatex-def
2197 \def\MT@tr@set@okern#1,#2,{%
2198   \let\MT@temp@empty
2199   \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
2200   \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
2201   \MT@glet@nc{\MT@outer@kern\expandafter\string\font@name}\MT@temp
2202   <debug>\MT@dinfo@n12{... outer kerning: (#1,#2)
2203   <debug> = \@nameuse{\MT@outer@kern\expandafter\string\font@name}}%
2204 }

\MT@tr@set@okern@

2205 \def\MT@tr@set@okern@#1{%
2206   \MT@test@ast#1*\@nil{%
2207     \MT@ifdefined@c@TF\MT@tr@unit@
2208     {\edef@tempb{#1}\MT@scale@to@em}
2209     {\@tempcntb=#1\relax}%

```

```

2210 \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
2211 }{%
2212 \MT@ifempty\@tempa{\let\@tempa\@m}\relax
2213 \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
2214 * \fontdimen6\MT@font/2000\relax
2215 }%
2216 \advance\@tempdima -\dimexpr \MT@letterspace@ sp
2217 * \fontdimen6\MT@font/2000\relax
2218 \edef\MT@temp{\MT@temp{\the\@tempdima}}%
2219 }
2220 (/pdfTeX-def|luaTeX-def)

```

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

```

2221 (*pdfTeX-def|luaTeX-def|letterspace)
2222 \def\MT@outer@k{%
2223 \ifhmode
2224 \ifdim\lastkern=-3sp \unkern
2225 \ifdim\lastkern=3sp \kern-3sp
2226 \expandafter\expandafter\expandafter\@gobble
2227 \else \unkern
2228 \expandafter\expandafter\expandafter\@firstofone
2229 \fi
2230 \else
2231 \expandafter\@firstofone
2232 \fi
2233 {\kern\MT@outer@kern\kern3sp\kern-3sp\relax}%
2234 \fi
2235 }
2236 (/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.

```

2237 (*pdfTeX-def|luaTeX-def)
2238 (pdfTeX-def)\MT@requires@pdfTeX5{
2239 \def\MT@noligatures{%
2240 \MT@dotrue
2241 \let\@tempa\MT@n1@setname
2242 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
2243 \MT@ifdefined@nTF{\MT@checklist@##1}%
2244 {\csname \MT@checklist@##1\endcsname}%
2245 {\MT@checklist@##1}}%
2246 {n1}%
2247 }%
2248 \ifMT@do
2249 \MT@noligatures@\MT@font\MT@n1@ligatures
2250 \fi
2251 }

```

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

```

2252 (luaTeX-def)\MT@requires@luaTeX4{\let\pdfnoligatures\ignoreligaturesinfont}\relax
2253 \def\MT@noligatures@#1#2{%
2254 \MT@ifdefined@c@TF#2%

```

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

```

2255 \MT@ifdefined@c@TF\tagcode%

```

No 'inputenc' key.

```

2256 \let\MT@warn@maybe@inputenc@empty
2257 \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
2258 \MT@map@clist@c#2%

```

```

2259      \KV@esp@def\@tempa{#1}\MT@get@slot
2260      \ifnum\MT@char>\m@ne
2261      \tagcode#1\MT@char=\m@ne

```

With LuaT<sub>E</sub>X, we additionally register the ligatures that should be inhibited in a table (used by the luaotfload function `keepligature`).

```

2262 \luatex-def      \MT@if@fontspec@font
2263 \luatex-def      {\MT@lua{microtype.noligatures([[#1]],[[\MT@char]])}}\relax
2264      \fi
2265      }%
2266      \MT@vinfo{... Disabling ligatures for characters: #2}%
2267      }%
2268      \pdfnoligatures#1%
2269      \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
2270      know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
2271      the font instead}%
2272      }%
2273      }%
2274      \pdfnoligatures#1%
2275 \luatex-def      \MT@if@fontspec@font
2276 \luatex-def      {\MT@lua{microtype.noligatures([[#1]],"_all_")}}\relax
2277      \MT@vinfo{... Disabling all ligatures}%
2278      }%
2279      }
2280 \pdfTEX-def}\relax
2281 \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@`.

```

2282 \luafile
2283 microtype.ligs = microtype.ligs or { }
2284
2285 local function noligatures(fontcs,liga)
2286   local fontcs = match(fontcs,"([^\ ]+)")
2287   microtype.ligs[fontcs] = microtype.ligs[fontcs] or { }
2288   table.insert(microtype.ligs[fontcs],liga)
2289 end
2290 microtype.noligatures = noligatures
2291
2292 local function keepligature(c)
2293   local nodedirect = node.direct
2294   local getfield   = nodedirect.getfield
2295   local getfont    = nodedirect.getfont
2296   local f,ch
2297   if type(c) == "userdata" then -- in older luaotfload versions, c was a node
2298     f = c.font
2299     ch = c.components.char
2300   else
2301     f = getfont(c)
2302     ch = getfield(getfield(c,"components"),"char")
2303   end
2304   -- if ch then -- should always be true
2305   local ligs = microtype.ligs[match(tex.fontidentifier(f),"\([^\ ]+)")]
2306   if ligs then
2307     for _,lig in pairs(ligs) do
2308       if lig == "_all_" or tonumber(lig) == ch then
2309         return false
2310       end
2311     end
2312   end
2313   return true
2314 -- end

```

```

2315 end
2316
2317 if luaotfload and luaotfload.letterspace then
2318   if luaotfload.letterspace.keepligature then
2319     microtype.warning("overwriting function `keepligature'")
2320   end
2321   luaotfload.letterspace.keepligature = keepligature
2322 end
2323
2324 (luafile)

```

### 14.2.7 Loading the configuration

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

```

2325 (*package)
2326 \def\MT@load@list#1{%
2327   \edef\@tempa{#1}%
2328   \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa @load}%
2329   \MT@ifstreq\@tempa\@tempb{%
2330     \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempa' cannot load itself}{}%
2331   }{%
2332     \ifx\@tempb\relax \else
2333       \MT@ifdefined@n@TF{MT@MT@feat @c@\@tempb}{%
2334         \MT@vinfo{... : First loading \@nameuse{MT@abbr@MT@feat} list `@\@tempb'}%
2335         \begin@group
2336           \MT@load@list\@tempb
2337         \end@group
2338         \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list
2339           \noexpand\MessageBreak`@\@tempb'}%
2340         \MT@let@cn\@tempc{MT@MT@feat @c@\@tempb}%
2341         \expandafter\MT@set@codes\@tempc,\relax,%
2342       }{%
2343         \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempb' undefined.\MessageBreak
2344           Cannot load it from list `@\@tempa'}{}%
2345       }%
2346     \fi
2347   }%
2348 }

```

`\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.

```

2349 \let\MT@file@list\@empty
2350 \def\MT@find@file#1{%
2351
2352   Check for existence of the file only once.
2353
2354   \MT@in@clist{#1}\MT@file@list
2355   \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.

```

2353   \MT@begin@catcodes
2354   \let\MT@begin@catcodes\relax
2355   \let\MT@end@catcodes\relax
2356   \InputIfFileExists{mt-#1.cfg}{%
2357     \edef\MT@curr@file{mt-#1.cfg}%
2358     \MT@vinfo{... Loading configuration file \MT@curr@file}%
2359     \MT@xadd\MT@file@list{#1,}%
2360   }{%
2361     \MT@get@basefamily#1\@empty\@empty\@empty\@nil
2362     \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2363     \ifMT@inlist@
2364       \MT@xadd\MT@file@list{#1,}%
2365     \else

```

```

2366     \InputIfFileExists{mt-\@tempa.cfg}{%
2367     \edef\MT@curr@file{mt-\@tempa.cfg}%
2368     \MT@vinfo{... Loading configuration file \MT@curr@file}%
2369     \MT@xadd\MT@file@list{\@tempa,#1,}%
2370     }{%
2371     \MT@vinfo{... No configuration file mt-#1.cfg}%
2372     \MT@xadd\MT@file@list{#1,}%
2373     }%
2374     \fi
2375     }%
2376     \endgroup
2377     \fi
2378 }

```

`\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.

```

2379 \def\MT@cfg@catcodes{%
2380   \makeatletter
2381   \catcode\^7%
2382   \catcode\ 9%
2383   \catcode\^^I9%
2384   \catcode\^^M9%
2385   \catcode\\\z@
2386   \catcode\{\@ne
2387   \catcode\}\@tw@
2388   \catcode\#6%
2389   \catcode\%14%
2390   \MT@map@tlist@n
2391   {\!"#$%&'(\)*+,\-.\/:;|<=>?[\]_`{|}~}%
2392   \@makeother
2393 }

```

`\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.

```

2394 \def\MT@begin@catcodes{%
2395   \begingroup
2396   \MT@cfg@catcodes
2397 }

```

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

```

2398 \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 `cmsx` and `cmsj` (OK, `cmex` will still become `cme` ...).

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

```

2399 \def\MT@get@basefamily#1#2#3#4@nil{%
2400   \ifx\@empty#4%
2401     \def\@tempa{#1#2#3}%
2402   \else
2403     \let\@tempa\@empty
2404     \edef\@tempb{#1#2#3#4}%
2405     \expandafter\MT@get@basefamily\@tempb\@nil
2406   \fi
2407 }

```

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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	-

`\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’.

```
2408 \def\MT@get@basefamily@#1#2\nil{%
2409   \edef\@tempa{\@tempa#1}%
2410   \ifx\#2\expandafter@gobble\else\expandafter\@firstofone\fi
2411   {\MT@in@tlist{#2}\MT@variants
2412    \ifMT@inlist\else\MT@get@basefamily@#2\nil\fi}%
2413 }
```

`\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@ 2414 \def\MT@get@listname#1{%
2415   (debug)\MT@dinfol{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
2416   \let\MT@listname\undefined
2417   \def\@tempb{#1}%
2418   \MT@map@tlist@c\MT@try@order\MT@get@listname@
2419 }
2420 \def\MT@get@listname@#1{%
2421   \expandafter\MT@next@listname#1%
2422   \ifx\MT@listname\undefined \else
2423     \expandafter\MT@tlist@break
2424     \fi
2425 }
```

`\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.

```
2426 \def\MT@try@order{%
2427   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2428   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2429 }
```

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

```
2430 \def\MT@next@listname#1#2#3#4{%
2431   \ifnum#1=\z@\MT@nofamilytrue\fi
2432   \edef\@tempa{\MT@encoding
2433    /\ifnum#1=\@ne \MT@family \fi
2434    /\ifnum#2=\@ne \MT@series \fi
2435    /\ifnum#3=\@ne \MT@shape \fi
2436    /\ifnum#4=\@ne *\fi
2437     \MT@context}%
2438   (debug)\MT@dinfol{1}{trying \@tempa}%
2439   \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
2440     \MT@next@listname@#4%
2441   }%}
```

Also try with an alias family.

```
2442   \ifnum#1=\@ne
2443     \ifx\MT@familyalias@empty \else
2444       \edef\@tempa{\MT@encoding
```

```

2445             /\MT@familyalias
2446             /\ifnum#2=\@ne \MT@series\fi
2447             /\ifnum#3=\@ne \MT@shape\fi
2448             /\ifnum#4=\@ne *\fi
2449             \MT@context}%
2450 <debug>\MT@info@n1{1}{(alias) \@tempa}%
2451             \MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%
2452             \MT@next@listname@#4%
2453             }%
2454             \fi
2455             \fi
2456             }%
2457 }

```

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

```

2458 \def\MT@next@listname@#1{%
2459   \ifnum#1=\@ne
2460     \MT@exp@cs\MT@in@rlist{MT@\@tempb @\@tempa @sizes}%
2461     \ifMT@inlist@
2462       \let\MT@listname\MT@size@name
2463     \fi
2464   \else
2465     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
2466   \fi
2467 }

```

`\MT@if@list@exists`

```

\MT@context 2468 \def\MT@if@list@exists{%
2469   \MT@let@cn\MT@context{MT@\MT@feat @context}%
2470   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
2471   \MT@get@listname{\MT@feat @c}%
2472   \MT@ifdefined@c@TF\MT@listname{%
2473     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
2474     \ifMT@nonselected
2475       \MT@vinfo{... Applying non-selected expansion (list `'\MT@listname')}%
2476     \else
2477       \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `'\MT@listname'}%
2478     \fi
2479     \@firstoftwo
2480   }%

```

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

```

2481   \MT@let@nc{MT@\MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

2482   \ifMT@nonselected
2483     \MT@vinfo{... Applying non-selected expansion (no list)}%
2484   \else

```

Tracking doesn't require a list, either.

```

2485   \MT@ifstreq\MT@feat{tr}\relax%
2486   \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
2487     for font\MessageBreak`'\MT@font'%
2488     \ifx\MT@context\@empty\else\space(context: `'\MT@context')\fi.
2489     Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
2490   }%
2491   \fi
2492   \@secondoftwo
2493 }%
2494 }

```

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

```

\MT@context 2495 \def\MT@get@inh@list{%
2496   \let\MT@context\@empty

```

```

2497 \MT@get@listname{\MT@feat @inh}%
2498 \MT@ifdefined@c@TF\MT@listname{%
2499 \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
2500 <debug>\MT@info@n1{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
2501 <debug> \MT@listname'}%
2502 \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%

```

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

```

2503 \ifx\@tempc\@empty \else
2504 <debug>\MT@info@n1{1}{parsing inheritance list ...}%

```

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

```

2505 \begingroup
2506 \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\MT@listname'}%
2507 \MT@set@inputenc{inh}%
2508 \expandafter\MT@inh@do\@tempc,\relax,%
2509 \MT@gl@et@nc{MT@\MT@feat @inh@\MT@listname}\@empty
2510 \endgroup
2511 \fi
2512 }%
2513 \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
2514 }%
2515 }

```

### 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@ 2516 \def\MT@get@slot{%
2517 \escapechar`\\
2518 \let\MT@char@\m@ne
2519 \MT@norestrtrue

```

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

```

2520 \MT@toks=\expandafter{\@tempa}%

```

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

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

```

2521 \expandafter\MT@is@letter\@tempa\relax\relax
2522 \ifnum\MT@char@ < \z@

```

- 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.

```

2523 \MT@exp@two@c\MT@is@active\string\@tempa\@nil

```

- 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.

```

2524 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%

```

```
2525 \MT@is@symbol
```

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

```
2526 {\expandafter\MT@is@composite\@tempa\relax\relax}%
2527 \ifnum\MT@char@ < \z@
```

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

```
2528 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2529 \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2530 \fi
2531 \fi

2532 \let\MT@char\MT@char@
2533 \MT@get@slot@
2534 \escapechar\m@ne
2535 }
2536 </package>
```

```
\MT@get@slot@
```

```
2537 <*\pdfTeX-def|luatex-def|xetex-def>
2538 \def\MT@get@slot@{%
```

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

```
2539 <xetex-def> \ifnum\XeTeXfonttype\MT@font=\z@
2540 \ifnum\MT@char > \m@ne
```

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

```
2541 <*\luatex-def>
2542 \ifnum\MT@char=47\relax
2543 \ifMT@noest \else
2544 \@tempcnta=\MT@lua{
2545 local glyph = microtype.name_to_slot([[ \expandafter\@gobble\@tempa ]], true)
2546 if glyph then tex.write(glyph)
2547 else tex.write(-1)
2548 end
2549 }\relax
2550 \ifnum\@tempcnta<\z@
2551 \MT@warn@unknown
2552 \let\MT@char\m@ne
2553 \else
2554 \edef\MT@char{\the\@tempcnta}%
2555 <debug>\MT@dinfol{3}{> `the\MT@toks' is a glyph name (\the\@tempcnta)}%
2556 \fi
2557 \fi
2558 \else
2559 </luatex-def>
```

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

```
2560 \ifMT@noest \else
2561 \MT@warn@rest
2562 <pdfTeX-def|luatex-def> \let\MT@char\m@ne
2563 <xetex-def> \let\MT@char\@empty
2564 \fi
2565 <luatex-def> \fi
2566 \else
2567 \MT@warn@unknown
2568 <xetex-def> \let\MT@char\@empty
2569 \fi
2570 <*\xetex-def>
2571 \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@.

```

2572 \ifnum\MT@char=47\relax
2573 \ifMT@noreset \edef\MT@char{U47}%
2574 \else
2575 \@tempcnta=\XeTeXglyphindex"\expandafter\@gobble\@tempa"\relax
2576 \ifnum\@tempcnta=\z@
2577 \MT@warn@unknown
2578 \let\MT@char\@empty
2579 \else
2580 \edef\MT@char{\@tempa\space}%
2581 \edef\MT@char@{-\the\@tempcnta}%
2582 (debug)\MT@dinfol{3}{> `-\the\MT@toks' is a glyph name (\the\@tempcnta)}%
2583 \fi
2584 \fi
2585 \else
2586 \ifnum\MT@char > \m@ne
2587 \ifMT@noreset

```

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

```

2588 \@tempcnta=\XeTeXcharglyph\MT@char\relax
2589 \ifnum\@tempcnta=\z@
2590 \MT@info@missing@char
2591 \let\MT@char\@empty
2592 \else
2593 (debug)\MT@dinfol{3}{> (glyph number: \the\@tempcnta,
2594 (debug) glyph name: \XeTeXglyphname\MT@font\@tempcnta)}%
2595 \edef\MT@char{U\MT@char}%
2596 \fi
2597 \else
2598 \MT@warn@rest
2599 \let\MT@char\@empty
2600 \fi
2601 \else
2602 \MT@warn@unknown
2603 \let\MT@char\@empty
2604 \fi
2605 \fi
2606 \fi
2607 (/xetex-def)
2608 }
2609 (/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.

```

2610 (*luafile)
2611 if luaotfload and luaotfload.aux and luaotfload.aux.slot_of_name then
2612 local slot_of_name = luaotfload.aux.slot_of_name
2613 microtype.name_to_slot = function(name, unsafe)
2614 return slot_of_name(font.current(), name, unsafe)
2615 end
2616 else
2617 -- we dig into internal structure (should be avoided)
2618 local function name_to_slot(name, unsafe)
2619 if fonts then
2620 local unicodes
2621 if fonts.ids then --- legacy luaotfload
2622 local tfmdata = fonts.ids[font.current()]
2623 if not tfmdata then return end
2624 unicodes = tfmdata.shared.otfdata.luatex.unicodes
2625 else --- new location
2626 local tfmdata = fonts.hashes.identifiers[font.current()]

```

```

2627     if not tfmdata then return end
2628     unicodes = tfmdata.resources.unicodes
2629     end
2630     local unicode = unicodes[name]
2631     if unicode then --- does the 'or' branch actually exist?
2632         return type(unicode) == "number" and unicode or unicode[1]
2633     end
2634     end
2635     end
2636     microtype.name_to_slot = name_to_slot
2637 end
2638
2639 </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 2640 <*pdfTeX-def|luaTeX-def|xetex-def>
2641 \def\MT@max@char
2642 <pdfTeX-def> {127 }
2643 <luaTeX-def|xetex-def> {1114111 }
2644 \def\MT@max@slot
2645 <pdfTeX-def> {255 }
2646 <luaTeX-def|xetex-def> {1114111 }
2647 </pdfTeX-def|luaTeX-def|xetex-def>

```

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

```

2648 <*package>
2649 \newif\ifMT@noest

2650 \def\MT@is@letter#1#2\relax{%
2651   \ifcat a\noexpand#1\relax
2652     \edef\MT@char@{\number`#1}%
2653     \ifx\#2\%
2654 <debug>\MT@dinfol{3}> `the\MT@toks' is a letter (\MT@char@)%
2655     \else
2656       \MT@noestfalse
2657     \fi
2658   \else
2659     \ifcat !\noexpand#1\relax
2660       \edef\MT@char@{\number`#1}%
2661 <debug>\MT@dinfol{3}> `the\MT@toks' is a character (\MT@char@)%
2662     \ifx\#2\%
2663       \ifnum\MT@char@ > \MT@max@char \MT@warn@ascii \fi
2664     \else
2665       \MT@noestfalse
2666       \expandafter\MT@is@number#1#2\relax\relax
2667     \fi
2668   \fi
2669 \fi
2670 }

```

`\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.

```

2671 \def\MT@is@number#1#2#3\relax{%
2672   \ifx\relax#3\relax \else
2673     \ifx\relax#2\relax \else
2674       \MT@noesttrue
2675       \if#1"\relax
2676         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2677 <debug>\MT@dinfol{3}> ... a hexadecimal number: \MT@char@%
2678       \else
2679         \if#1'\relax

```

```

2680     \def\MT@char@{\number#1#2#3}%
2681 (debug)\MT@info@n1{3}{> ... an octal number: \MT@char@}%
2682     \else
2683     \MT@ifint{#1#2#3}{%
2684     \def\MT@char@{\number#1#2#3}%
2685 (debug)\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
2686     }\MT@noestfalse
2687     \fi
2688     \fi
2689     \ifnum\MT@char@ > \MT@max@slot
2690     \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2691     \let\MT@char@\m@ne
2692     \fi
2693     \fi
2694     \fi
2695 }

```

`\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.

```

2696 \def\MT@is@active#1#2\@nil{%
2697   \ifnum\catcode`#1 = \active
2698   \begingroup
2699   \set@display@protect
2700   \let\IeC\@firstofone
2701   \let\@inpenc@undefined@\MT@undefined@char

```

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

```

2702   \def\UTFviii@defined##1{\ifx ##1\relax
2703   \MT@undefined@char{utf8}\else\expandafter ##1\fi}%

```

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

```

2704   \MT@ifdefined@cT\PrerenderUnicode
2705   {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
2706   \edef\x{\endgroup
2707   \def\noexpand\@tempa{\@tempa}%

```

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

```

2708   \MT@toks={\the\MT@toks\space(= \@tempa)}%
2709   }%
2710   \x
2711   \fi
2712 }

```

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

```

2713 \def\MT@undefined@char#1{undefined in input encoding ``#1''}

```

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

```

2714 \def\MT@is@symbol{%
2715   \expandafter\def\expandafter\MT@char\expandafter
2716   {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2717   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter

```

```
2718 \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2719 \ifnum\MT@char@ < \z@
```

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

```
2720 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2721 \fi
2722 }
```

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

```
\MT@charstring 2723 \begingroup
2724 \catcode`\=\z@

2725 /MT@map@tlist@n{\CHARLEX}/@makeother
2726 /lowercase{%
2727 /def/x{/endgroup
2728 /def/MT@charstring{\CHAR"%
2729 /def/MT@is@char##1\CHAR"##2##3##4/relax{%
2730 /ifx/relax##4/relax
2731 /ifMT@xunicode
2732 /expandafter/MT@is@charx/MT@strip@prefix##1>/relax\CHAR "%
2733 /relax/relax/relax/relax/relax
2734 /fi
2735 /else
2736 /ifx/relax##1/relax
2737 /if##3\relax
2738 /edef/MT@char@{/number"##2}%
2739 /MT@ifstreq/MT@charstring{##3##4}/relax/MT@norestfalse
2740 /else
2741 /edef/MT@char@{/number"##2##3}%
2742 /MT@ifstreq/MT@charstring{##4}/relax
2743 {/MT@is@xchar##2##3|##4\CHAR"/relax}%
2744 /fi
2745 <debug> /MT@dinfo@n1{3}{>~/the/MT@toks' is a \char (/MT@char@)}%
2746 /fi
2747 /fi
2748 }%
```

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

```
2749 /def/MT@is@xchar##1|##2\CHAR"##3##4/relax{%
2750 /MT@ifstreq/MT@charstring{##3##4}%
2751 {/edef/MT@char@{/number"##1##2}}/MT@norestfalse
2752 }%
```

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

```
\MT@strip@prefix 2753 /def/MT@charxstring{\CHAR "%
2754 /def/MT@strip@prefix##1>##2/relax{##2}%
\MT@is@charx 2755 /def/MT@is@charx##1\CHAR "##2##3##4##5##6/relax{%
2756 /ifx/relax##1/relax
2757 /ifx/relax##6/relax/else
2758 /edef/MT@char@{/number"##2##3##4##5}%
2759 /MT@ifstreq{\RELAX >\CHAR "}{##6}/relax/MT@norestfalse
2760 <debug> /MT@dinfo@n1{3}{>~/the/MT@toks' is a xunicode \char (/MT@char@)}%
2761 /fi
2762 /fi
2763 }%
2764 }%
2765 }
2766 /x
```

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

```
2767 \def\MT@is@composite#1#2\relax{%
2768 \ifx\#2\\\else
```

Again, we construct a control sequence, this time of the form: `\\<encoding>`

$\backslash\langle accent\rangle\langle character\rangle$ , e.g.,  $\backslash\text{T1}\langle -a\rangle$ , which we then expand once to see if it is a letter (if it has been defined by  $\backslash\text{DeclareTextComposite}$ ). This should be robust, finally, especially, since we also  $\backslash\text{detokenize}$  the input instead of only  $\backslash\text{stringify}$  it. Thus, we will die gracefully even on wrong Unicode input without utf8.

```
2769 \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2770 \string\csname\MT@encoding\endcsname
2771 \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 ( $\backslash\text{DeclareUnicodeComposite}$ ), which we take care of here ( $\backslash\text{UnicodeEncodingName}$  has been introduced at the same time):

```
2772 \ifx\UnicodeEncodingName\undefined\else
2773 \expandafter\expandafter\expandafter
2774 \MT@is@uni@comp\MT@char\iffontchar\else\fi\relax
2775 \fi
2776 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
```

Again, xunicode.

```
2777 \ifnum\MT@char@ < \z@
2778 \ifMT@xunicode
2779 \edef\MT@char{\MT@exp@two@c\MT@strip@prefix\meaning\MT@char>\relax}%
2780 \expandafter\MT@exp@two@c\expandafter\MT@is@charx\expandafter
2781 \MT@char\MT@charxstring\relax\relax\relax\relax\relax
2782 \fi
2783 \fi
2784 \fi
2785 }
```

$\text{MT@is@uni@comp}$  Helper for  $\backslash\text{DeclareUnicodeComposite}$ .

```
2786 \def\MT@is@uni@comp#1\iffontchar#2\else#3\fi\relax{%
2787 \ifx\#2\else\edef\MT@char{\iffontchar#2\fi}\fi
2788 }
```

[What about math? Well, for a moment the following looked like a solution, with  $\backslash\text{mt@is@mathchar}$  defined accordingly, analogous to  $\backslash\text{MT@is@char}$  above, to pick up the last two tokens (the  $\backslash\text{meaning}$  of a  $\backslash\text{mathchardef}$ '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  $\backslash\text{mathcodes}$  and  $\backslash\text{mathchardefs}$  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.

$\backslash\text{MT@curr@list@name}$  The type and name of the current list, defined at various places.

```
\MT@set@listname 2789 \def\MT@set@listname{%
2790 \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list\noexpand\MessageBreak
2791 \~\@nameuse{MT@MT@feat @c@name}'}%
2792 }
```

$\backslash\text{MT@warn@ascii}$  For 'other' characters > 127, we issue a warning (inputenc probably hasn't been

loaded), since correspondence with the slot numbers would be purely coincidental.

```
2793 \def\MT@warn@ascii{%
2794   \MT@warning@n1{Character `\'the\MT@toks' (= \MT@char@)
2795   is outside of ASCII range.\MessageBreak
2796   You must load the `inputenc' package before using\MessageBreak
2797   8-bit characters in \MT@curr@list@name}%
2798 }
```

\MT@warn@number@too@large     Number too large.

```
2799 \def\MT@warn@number@too@large#1{%
2800   \MT@warning@n1{%
2801     Number #1 in encoding `\'MT@encoding' too large!\MessageBreak
2802     Ignoring it in \MT@curr@list@name}%
2803 }
```

\MT@warn@rest     Not all of the string has been parsed.

```
2804 \def\MT@warn@rest{%
2805   \MT@warning@n1{%
2806     Unknown slot number of character\MessageBreak`\'the\MT@toks'%
2807     \MT@warn@maybe@inputenc\MessageBreak
2808     in font encoding `\'MT@encoding'.\MessageBreak
2809     Make sure it's a single character\MessageBreak
2810     (or a number) in \MT@curr@list@name}%
2811 }
```

\MT@warn@unknown     No idea what went wrong.

```
2812 \def\MT@warn@unknown{%
2813   \MT@warning@n1{%
2814     Unknown slot number of character\MessageBreak`\'the\MT@toks'%
2815     \MT@warn@maybe@inputenc\MessageBreak
2816     in font encoding `\'MT@encoding' in \MT@curr@list@name}%
2817 }
```

\MT@warn@maybe@inputenc     In case an input encoding had been requested.

```
2818 \def\MT@warn@maybe@inputenc{%
2819   \MT@ifdefined@n@T
2820   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
2821   { (input encoding `\'@nameuse
2822     {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2823 }
```

### 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`), `t ipa`, 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 2824 \let\MT@font@list\@empty
2825 \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`.

```
2826 </package>
2827 <+package|letterspace>
2828 <plain>\MT@requires@latex2{
2829 \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.

```
2830 <package> \MT@with@package@T{luatexja}{\MT@warn@unknown@once{luatexja}}%
2831 <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`.

```
2832 \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.

```
2833 \ifpackageloaded{xeCJK}{\@firstofone}{%
2834 \ifpackageafter{CJK}{2006/10/17}% 4.7.0
2835 {\def\MT@orig@pickupfont{\CJK@ifundefined\CJK@plane}}%
2836 {\def\MT@orig@pickupfont{\ifundefined{CJK@plane}}}%
2837 \g@addto@macro\MT@orig@pickupfont
2838 {{{\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).

```

2839 \ifpackageloaded{CJKutf8}%
2840   {\ifpackagelater{CJKutf8}{2008/05/22}% 4.8.0
2841     {\ifpdf\expandafter\secondoftwo\else\expandafter\firstoftwo\fi}%
2842     {\@firstoftwo}}%
2843     {\@firstoftwo}%
2844   {\g@addto@macro\MT@orig@pickupfont{%
2845     {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2846       \define@newfont\else\xdef\font@name{%
2847         \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
2848   {\g@addto@macro\MT@orig@pickupfont{%
2849     {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2850       \define@newfont\def\CJK@temp{v}%
2851       \ifx\CJK@temp\CJK@plane
2852         \expandafter\ifx\csname CJK@cmap@\f@family\CJK@plane\endcsname\relax
2853         \else\csname CJK@cmap@\f@family\CJK@plane\endcsname\fi
2854         \else \CJK@addcmap\CJK@plane \fi
2855         \else\xdef\font@name{%
2856           \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
2857   \@gobble
2858   }%
2859 }{\@firstofone}%

```

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

```

2860 {\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.

```

2861 \ifx\pickup@font\MT@orig@pickupfont \else
2862   \MT@warning@nl{%
2863     Command \string\pickup@font\space is not defined as expected.%
2864     \MessageBreak Patching it anyway. Some things may break%
2865   }{*package}
2866     .\MessageBreak Double-check whether micro-typography is indeed%
2867     \MessageBreak applied to the document.%
2868     \MessageBreak (Hint: Turn on `verbose' mode)%
2869   }{/package}
2870   }%
2871   \fi

```

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

```

2872 \g@addto@macro\pickup@font{\begingroup}%

```

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

```

2873 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
2874 \g@addto@macro\pickup@font{%
2875   \escapechar\m@ne
2876 }{*package}
2877 {debug} \global\MT@inannottrue
2878 {debug} \MT@gl@et\MT@pdf@annot\@empty
2879 {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.

```

2880 \MT@let@cn\MT@font\MT@subst\expandafter\string\font@name}%
2881 \ifx\MT@font\relax
2882   \let\MT@font\font@name
2883 \else
2884   \ifx\MT@font\font@name \else

```

```

2885 <debug> \MT@addto@annot{= substituted with \MT@font}%
2886         \MT@register@subst@font
2887         \fi
2888         \fi
2889         \MT@setupfont
2890 </package>
2891 <letterspace> \MT@tracking
2892         \endgroup
2893     }%
2894 <*package>

```

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

```

\MT@MT@pickupfont
\MT@ltx@pickupfont 2895 \let\MT@pickupfont\pickup@font
2896 \def\MT@MT@pickupfont {\let\pickup@font\MT@pickupfont}%
2897 \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.

```

2898 \g@addto@macro\do@subst@correction
2899 { \edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2900   \MT@glet@nc{MT@subst@expandafter\string\font@name}\MT@font}%

```

\add@accent Inside \add@accent, we have to disable microtype's setup, since the grouping in the patched \pickup@font would break the accent if different fonts are used for the base character and the accent. Fortunately, L<sup>A</sup>T<sub>E</sub>X takes care that the fonts used for the \accent are already set up, so that we cannot be overlooking them.

```

2901 \let\MT@orig@add@accent\add@accent
2902 \def\add@accent#1#2{%
2903   \MT@ltx@pickupfont
2904   \MT@orig@add@accent{#1}{#2}%
2905   \MT@MT@pickupfont
2906 }%
2907 </package>
2908 }
2909 <plain>\relax
2910 <*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.

```

2911 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}

```

\MT@register@font Register the current font.

```

2912 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}

```

\MT@register@subst@font Register the substituted font (only if isn't registered already).

```

2913 \def\MT@register@subst@font{\MT@exp@one@n\MT@in@clist\font@name\MT@font@list
2914   \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.

```

2915 \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.

```

2916 \def\MT@check@font@cx{%
2917   \MT@if@true
2918   \MT@map@clist@c\MT@active@features{%
2919     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
2920     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2921     \ifMT@inlist@
2922       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
2923     \else
2924       \MT@if@false
2925     \fi
2926   }%
2927   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2928 }

```

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

```

2929 \def\MT@register@subst@font@cx{%
2930   \MT@map@clist@c\MT@active@features{%
2931     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name
2932     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2933     \ifMT@inlist@ \else
2934       \MT@exp@cs\MT@xadd
2935       {MT@##1@\csname MT@##1@context\endcsname font@list}%
2936       {\font@name,}%
2937     \fi
2938   }%
2939 }

```

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

```

2940 \def\MT@register@font@cx{%
2941   \MT@map@clist@c\MT@active@features{%
2942     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
2943     \MT@exp@cs\MT@xadd
2944     {MT@##1@\csname MT@##1@context\endcsname font@list}%
2945     {\MT@font,}%
2946     \def\@tempa{##1}%
2947     \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
2948     \fi
2949   }%
2950 }

```

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

```

2951 \def\MT@maybe@rem@from@list#1{%
2952   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
2953     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
2954     \MT@font \csname MT@\@tempa @#1font@list\endcsname
2955   }%
2956 }

```

`\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.

```

2957 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}
2958 \MT@addto@setup{%
2959   \DeclareRobustCommand\microtypecontext[1]{%
2960     \MT@setup@contexts
2961     \let\MT@reset@context\relax

```

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

```

2962 \MT@glet@glb@currsizel@empty
2963 \setkeys{MTC}{#1}%
2964 \selectfont
2965 \MT@reset@context
2966 }%
2967 }

```

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

```

2968 \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@`

```

2969 \def\MT@reset@context@{%
2970 \MT@vinfo{<<< Resetting contexts\on@line
2971 <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
2972 <debug> \MT@tr@context/\MT@kn@context/\MT@sp@context
2973 }%
2974 \selectfont
2975 }

```

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

```

2976 \def\MT@setup@contexts{%
2977 \MT@map@clist@c\MT@active@features
2978 {\MT@glet@nc{MT@##1@font@list}\MT@font@list}%
2979 \MT@glet\MT@check@font\MT@check@font@cx
2980 \MT@glet\MT@register@font\MT@register@font@cx
2981 \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
2982 \MT@glet\MT@setup@contexts\relax
2983 }

```

Define context keys.

```

2984 \MT@map@clist@c\MT@features@long{%
2985 \define@key{MTC}{#1}[]{%
2986 \edef\@tempb{\@nameuse{MT@rbba@#1}}%
2987 \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2988 \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).

```

2989 \MT@ifempty{##1}{\def\MT@val{0}}{\def\MT@val{##1}}%
2990 \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
2991 <debug> \MT@dinfo{1}{>>> no change of #1 context: `'\MT@val'}%
2992 \else
2993 \MT@vinfo{>>> Changing #1 context to `'\MT@val'\MessageBreak\on@line
2994 <debug> \space(previous: `'\@nameuse{MT@\@tempb @context}')%
2995 }%
2996 \def\MT@reset@context{\aftergroup\MT@reset@context@}%

```

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

```

2997 \MT@glet@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes}%

```

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

```

2998 \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
2999 \MT@val \csname MT@\@tempb @doc@contexts\endcsname
3000 \ifMT@inlist@ \else
3001 \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{\MT@val}%
3002 <debug> \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
3003 \fi
3004 \MT@edef@n{MT@\@tempb @context}{\MT@val}%
3005 \fi
3006 \fi
3007 }%
3008 }

```

We also allow the activate shortcut.

```
3009 \define@key{MTC}{activate}[]{%
3010   \setkeys{MT}{protrusion=#1}}%
3011   \setkeys{MT}{expansion=#1}}%
3012 }
```

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

```
\MT@ex@context 3013 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
\MT@tr@context 3014 \MT@def@n{MT@#1@context}{@}%
\MT@sp@context 3015 \MT@def@n{MT@#1@doc@contexts}{@}%
\MT@kn@context 3016 }
```

```
\MT@tr@context 3014 \MT@def@n{MT@#1@context}{@}%
\MT@sp@context 3015 \MT@def@n{MT@#1@doc@contexts}{@}%
\MT@kn@context 3016 }
```

```
\MT@sp@context 3016 }
```

```
\MT@kn@context 3017 \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.

```
3018 \def\DeclareMicrotypeSet{%
3019   \MT@begin@catcodes
3020   \ifstar
3021     \MT@DeclareSetAndUseIt
3022     \MT@DeclareSet
3023 }
```

```
\MT@DeclareSet
```

```
3024 \newcommand\MT@DeclareSet[3][]{%
3025   \MT@ifempty{#1}{%
3026     \MT@map@clist@c\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%
3027   }%
3028   \MT@map@clist@n{#1}{%
3029     \MT@ifempty{#1}\relax{%
3030       \MT@is@feature{##1}{set declaration `#2'}{%
3031         \MT@exp@one@n\MT@declare@sets
3032         {\csname MT@rba@##1\endcsname}{#2}{#3}%
3033       }%
3034     }%
3035   }%
3036 }%
3037 \MT@end@catcodes
3038 }
```

```
\MT@DeclareSetAndUseIt
```

```
3039 \newcommand\MT@DeclareSetAndUseIt[3][]{%
3040   \MT@DeclareSet[#1]{#2}{#3}%
3041   \UseMicrotypeSet[#1]{#2}%
3042 }
```

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

```
3043 \let\MT@curr@set@name\empty
```

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

```
3044 \def\MT@declare@sets#1#2#3{%
3045   \def\MT@curr@set@name{#2}%
3046   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
3047     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set ` \MT@curr@set@name' }%
3048   }
```

```

3048 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
3049 \MT@gl@et@nc{MT@#1list@##1@MT@curr@set@name}\@undefined
3050 }%
3051 }%
3052 \MT@gl@et@nc{MT@#1set@MT@curr@set@name}\@empty
3053 <debug>\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
3054 \setkeys{MT@#1set}{#3}%
3055 }

```

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

```

3056 \def\MT@define@set@key@#1#2{%
3057 \define@key{MT@#2set}{#1}[]{}%
3058 \MT@gl@et@nc{MT@#2list@#1@MT@curr@set@name}\@empty
3059 \MT@map@clist@n{##1}{%
3060 \KV@sp@def\MT@val{###1}%
3061 \MT@get@highlevel{#1}%

```

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

```

3062 \MT@exp@two@n@g@addto@macro
3063 {\csname MT@#2list@#1@MT@curr@set@name\expandafter\endcsname}%
3064 {\MT@val,}%
3065 }%

```

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

```

3066 \expandafter\g@addto@macro\expandafter\MT@font@sets
3067 \csname MT@#2list@#1@MT@curr@set@name\endcsname
3068 <debug>\MT@dinfo@n1{1}{-- #1: \@nameuse{MT@#2list@#1@MT@curr@set@name}}%
3069 }%
3070 }

```

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

```

3071 \def\MT@get@highlevel#1{%
3072 \expandafter\MT@test@ast\MT@val*\@nil\relax}%

```

And ‘family = \*’ will become \familydefault.

```

3073 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
3074 \edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}%

```

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

```

3075 }%
3076 }

```

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

```

3077 \def\MT@test@ast#1*#2\@nil{%
3078 \def\@tempa{#1}%
3079 \MT@ifempty{#2}%
3080 }

```

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

```

3081 \let\MT@font@sets\@empty
3082 \def\MT@fix@font@set#1{%
3083 \MT@ifdefined@c@T{#1}{%
3084 \xdef#1{#1}%
3085 \ifMT@fontspec
3086 \xdef#1{\expandafter\MT@scrubfeatures#1()\relax}%
3087 \fi
3088 \global\@onelevel@sanitize#1%
3089 }%
3090 }

```

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

```

3091 \def\MT@define@set@key@size#1{%
3092   \define@key{MT@#1@set}{size}[]{%
3093     \MT@map@cliston{##1}%
3094     \def\MT@val{###1}%
3095     \expandafter\MT@get@range\MT@val--\@nil
3096     \ifx\MT@val\relax \else
3097       \MT@exp@cs\MT@xadd
3098       {MT@#1list@size@MT@curr@set@name}%
3099       {{{\MT@lower}{\MT@upper}\relax}}%
3100     \fi
3101   }%
3102   <debug>\MT@din@fo@n1{1}{-- size: \@nameuse{MT@#1list@size@MT@curr@set@name}}%
3103   }%
3104 }

```

Font sizes may also be specified as ranges. This has been requested by Andreas Bhmann, 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 is trying to do this for the OpenType version of Adobe's Minion. See <http://developer.berlios.de/projects/minionpro/>.)

`\MT@get@range` Ranges will be stored as triplets of `{\lower bound}{\upper bound}{\list name}`.  
`\MT@upper` For simple sizes, the upper boundary is `-1`.

```

\MT@lower 3105 \def\MT@get@range#1-#2-#3\@nil{%
3106   \MT@ifempty{#1}%
3107   \MT@ifempty{#2}%
3108   \let\MT@val\relax
3109   }%
3110   \def\MT@lower{0}%
3111   \def\MT@val{#2}%
3112   \MT@get@size
3113   \edef\MT@upper{\MT@val}%
3114   }%
3115   }%
3116   \def\MT@val{#1}%
3117   \MT@get@size
3118   \ifx\MT@val\relax \else
3119     \edef\MT@lower{\MT@val}%
3120     \MT@ifempty{#2}%
3121     \MT@ifempty{#3}%
3122     {\def\MT@upper{-1}}%

```

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

```

3123   {\def\MT@upper{2048}}%
3124   }%
3125   \def\MT@val{#2}%
3126   \MT@get@size
3127   \ifx\MT@val\relax \else
3128     \MT@ifdim\MT@lower>\MT@val{%
3129       \MT@error{%
3130         Invalid size range (\MT@lower\space > \MT@val) in font set
3131         ~\MT@curr@set@name'.\MessageBreak Swapping sizes}}%
3132     \edef\MT@upper{\MT@lower}%
3133     \edef\MT@lower{\MT@val}%
3134     }%
3135     \edef\MT@upper{\MT@val}%
3136     }%
3137     \MT@ifdim\MT@lower=\MT@upper
3138     {\def\MT@upper{-1}}%
3139     \relax
3140   \fi
3141   }%
3142   \fi

```

```
3143 }%
3144 }
```

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

```
3145 \def\MT@get@size{%
    A single star would mean \sizedefault, which doesn't exist, so we define it to be
    \normalsize.
3146 \if*\MT@val\relax
3147 \def\@tempa{\normalsize}%
3148 \else
3149 \MT@let@cn\@tempa{\MT@val}%
3150 \fi
3151 \ifx\@tempa\relax \else
```

The `resize` 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).

```
3152 \begingroup
3153 \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
3154 \@tempa\@nil
3155 \fi
```

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

```
3156 \MT@ifdimen\MT@val{%
3157 \@defaultunits\@tempdima\MT@val pt\relax\@nnil
3158 \edef\MT@val{\strip@pt\@tempdima}%
3159 }{%
3160 \MT@warning{Could not parse font size `~\MT@val'\MessageBreak
3161 in font set `~\MT@curr@set@name'}%
3162 \let\MT@val\relax
3163 }%
3164 }
```

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

```
3165 \def\MT@define@set@key@font#1{%
3166 \define@key{MT@#1@set}{font}[]{}%
3167 \MT@gl@et@nc{MT@#1list@font@\MT@curr@set@name}\@empty
3168 \MT@map@clist@n{##1}{%
3169 \def\MT@val{####1}%
3170 \MT@ifstreq\MT@val*{\def\MT@val{*/**/*/*}}\relax
3171 \expandafter\MT@get@font\MT@val////\@nil
3172 \MT@exp@two@n@g@addto@macro
3173 {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
3174 {\MT@val,}%
3175 }%
3176 \expandafter\g@addto@macro\expandafter\MT@font@sets
3177 \csname MT@#1list@font@\MT@curr@set@name\endcsname
3178 (debug)\MT@dinfo@n1{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
3179 }%
3180 }
```

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

```
3181 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
3182 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
3183 \ifx\MT@val\relax\def\MT@val{0}\fi
3184 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
3185 \let\MT@val\@tempb
3186 }
```

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

```

3187 \def\MT@get@font@#1#2#3#4#5#6{%
3188   \let\@tempb\@empty
3189   \def\MT@temp{#1/#2/#3/#4/#5}%
3190   \MT@get@axis{encoding}{#1}%
3191   \MT@get@axis{family}{#2}%
3192   \MT@get@axis{series}{#3}%
3193   \MT@get@axis{shape}{#4}%
3194   \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
3195   \MT@ifempty{#5}{%
3196     \MT@warn@axis@empty{size}{\string\normalsize}%
3197     \def\MT@val{*}%
3198   }{%
3199     \def\MT@val{#5}%
3200   }%
3201   \MT@get@size
3202 }

```

\MT@get@axis

```

3203 \def\MT@get@axis#1#2{%
3204   \def\MT@val{#2}%
3205   \MT@get@highlevel{#1}%
3206   \MT@ifempty\MT@val{%
3207     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
3208     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
3209   }\relax
3210   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
3211 }

```

\MT@warn@axis@empty

```

3212 \def\MT@warn@axis@empty#1#2{%
3213   \MT@warning{#1 axis is empty in font specification\MessageBreak
3214     ~\MT@temp'. Using ~#2' instead}%
3215 }

```

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

```

3216 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
3217   \MT@define@set@key@{encoding}{#1}%
3218   \MT@define@set@key@{family}{#1}%
3219   \MT@define@set@key@{series}{#1}%
3220   \MT@define@set@key@{shape}{#1}%
3221   \MT@define@set@key@size{#1}%
3222   \MT@define@set@key@font{#1}%
3223 }

```

\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.

```

3224 \def\UseMicrotypeSet%
3225   \MT@begin@catcodes
3226   \MT@UseMicrotypeSet
3227 }

```

\MT@UseMicrotypeSet

```

3228 \newcommand*\MT@UseMicrotypeSet[2][{}]{%
3229   \MT@ifempty{#1}{%
3230     \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
3231   }{%
3232     \MT@map@clist@n{#1}{%
3233       \MT@ifempty{##1}\relax%
3234       \MT@is@feature{##1}{activation of set ~#2'}%
3235       \MT@exp@one@n\MT@use@set
3236       {\csname MT@rbb@##1\endcsname}{#2}%
3237     }%
3238   }%

```

```

3239     }}%
3240   }%
3241   \MT@end@catcodes
3242 }

\MT@pr@setname    Only use sets that have been declared.
\MT@ex@setname 3243 \def\MT@use@set#1#2{%
\MT@tr@setname 3244 \MT@ifdefined@n@TF{MT@#1@set@@#2}{%
3245   \MT@xdef@n{MT@#1@setname}{#2}%
\MT@sp@setname 3246 }{%
\MT@kn@setname 3247 \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
\MT@use@set 3248   \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
3249   }%
3250   \MT@error{%
3251     The \@nameuse{MT@abbr@#1} set `#2' is undeclared.\MessageBreak
3252     Using set ` \@nameuse{MT@#1@setname}' instead}}%
3253   }%
3254 }

```

`\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.

```

3255 \def\DeclareMicrotypeSetDefault{%
3256   \MT@begin@catcodes
3257   \MT@DeclareMicrotypeSetDefault
3258 }

```

`\MT@DeclareMicrotypeSetDefault`

```

3259 \newcommand*\MT@DeclareMicrotypeSetDefault[2][ ]{%
3260   \MT@ifempty{#1}{%
3261     \MT@map@clistc\MT@features{{\MT@set@default@set{##1}{#2}}}%
3262   }{%
3263     \MT@map@clistn{#1}{%
3264       \MT@ifempty{#1}\relax%
3265       \MT@is@feature{##1}{declaration of default set `#2'}{%
3266         \MT@exp@one@n\MT@set@default@set
3267         {\csname MT@rbba@#1\endcsname}{#2}%
3268       }%
3269     }%
3270   }}%
3271 }%
3272 \MT@end@catcodes
3273 }

```

`\MT@default@pr@set`

```

\MT@default@ex@set 3274 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 3275 \MT@ifdefined@n@TF{MT@#1@set@@#2}{%
3276 <debug>\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set `#2'}%
\MT@default@sp@set 3277 \MT@xdef@n{MT@default@#1@set}{#2}%
\MT@default@kn@set 3278 }{%
\MT@set@default@set 3279 \MT@error{%
3280   The \@nameuse{MT@abbr@#1} set `#2' is not declared.\MessageBreak
3281   Cannot make it the default set. Using set\MessageBreak `all' instead}}%
3282   \MT@xdef@n{MT@default@#1@set}{all}%
3283   }%
3284 }

```

### 14.3.2 Variants and aliases

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

`\MT@variants`

```

3285 \let\MT@variants\@empty
3286 \def\DeclareMicrotypeVariants{%
3287   \MT@begin@catcodes

```

```

3288 \ifstar
3289 \MT@DeclareVariants
3290 {\let\MT@variants\empty\MT@DeclareVariants}%
3291 }

```

\MT@DeclareVariants

```

3292 \def\MT@DeclareVariants#1{%
3293 \MT@map@clist@n{#1}{%
3294 \def\@tempa{##1}%
3295 \@onelevel@sanitize\@tempa
3296 \xdef\MT@variants{\MT@variants{\@tempa}}%
3297 }%
3298 \MT@end@catcodes
3299 }

```

\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.

```

3300 \def\DeclareMicrotypeAlias{%
3301 \MT@begin@catcodes
3302 \MT@DeclareMicrotypeAlias
3303 }

```

\MT@DeclareMicrotypeAlias

```

3304 \newcommand*\MT@DeclareMicrotypeAlias[2]{%
3305 \def\@tempb{#2}%
3306 \@onelevel@sanitize\@tempb
3307 \MT@ifdefined@n@T{MT@#1@alias}{%
3308 \MT@warning{Alias font family \@tempb' will override
3309 alias \@nameuse{MT@#1@alias}'\MessageBreak
3310 for font family `#1'}}%
3311 \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.

```

3312 \MT@ifdefined@c@T\MT@family{%
3313 <debug>\MT@edinfo{1}{Activating alias font \@tempb' for \@MT@family'}%
3314 \MT@glet\MT@familyalias\@tempb
3315 }%
3316 \MT@end@catcodes
3317 }

```

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

```

3318 \def\LoadMicrotypeFile#1{%
3319 \edef\@tempa{\zap@space#1 \@empty}%
3320 \@onelevel@sanitize\@tempa
3321 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
3322 \ifMT@inlist@
3323 \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
3324 \else
3325 \MT@xadd\MT@file@list{\@tempa,}%
3326 \MT@begin@catcodes
3327 \InputIfFileExists{mt-\@tempa.cfg}{%
3328 \edef\MT@curr@file{mt-\@tempa.cfg}%
3329 \MT@vinfo{... Loading configuration file \MT@curr@file}%
3330 }{%
3331 \MT@warning{Configuration file mt-\@tempa.cfg\MessageBreak
3332 does not exist}%
3333 }%
3334 \MT@end@catcodes
3335 \fi
3336 }
3337 </package>
3338 </package|letterspace>

```

### 14.3.3 Disabling ligatures

`\DisableLigatures` This is really simple now: we can re-use the set definitions of `\DeclareMicrotypeSet`; there can only be one set, which we'll call 'no ligatures'.

`\MT@n1@setname` The optional argument may be used to disable selected ligatures only.

```

\MT@n1@ligatures 3339 <*pdftex-def|luatex-def
3340 <pdftex-def\MT@requires@pdftex5{
3341 \def\DisableLigatures{%
3342 \MT@begin@catcodes
3343 \MT@DisableLigatures
3344 }
3345 \newcommand*\MT@DisableLigatures[2] [] {%
3346 \MT@ifempty{#1}\relax{\gdef\MT@n1@ligatures{#1}}%
3347 \xdef\MT@active@features{\MT@active@features,n1}%
3348 \global\MT@noligaturestrue
3349 \MT@declare@sets{n1}{no ligatures}{#2}%
3350 \gdef\MT@n1@setname{no ligatures}%
3351 \MT@end@catcodes
3352 }
3353 <pdftex-def}{
3354 </pdftex-def|luatex-def

```

If pdf~~T~~~~E~~~~X~~ is too old, we throw an error.

```

3355 <*pdftex-def|xetex-def
3356 \renewcommand*\DisableLigatures[2] [] {%
3357 \MT@error{Disabling ligatures of a font is only possible\MessageBreak
3358 with pdftex version 1.30 or newer.\MessageBreak
3359 Ignoring \string\DisableLigatures}{%
3360 <pdftex-def Upgrade
3361 <xettex-def Use
3362 pdftex.}%
3363 }
3364 <pdftex-def}{
3365 </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.

```

3366 <*package>
3367 \def\DeclareMicrotypeBabelHook#1#2{%
3368 \MT@map@clist@n{#1}{%
3369 \KV@sp@def\@tempa{##1}%
3370 \MT@gdef@n{MT@babel@\@tempa}{#2}%
3371 }%
3372 }
3373 </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).

```

3374 <*pdftex-def|xetex-def|luatex-def
3375 \def\SetProtrusion{%
3376 \MT@begin@catcodes

```

```

3377 \MT@SetProtrusion
3378 }

\MT@SetProtrusion    We want the catcodes to be correct even if this is called in the preamble.
  \MT@pr@c@name 3379 \newcommand*\MT@SetProtrusion[3] [] {%
\MT@extra@context 3380 \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.
3381 \MT@set@named@keys{MT@pr@c}{#1}%
3382 debug\MT@dinfo{1}{creating protrusion list `~\MT@pr@c@name'}%
3383 \def\MT@permutelist{pr@c}%
3384 \setkeys{MT@cfg}{#2}%

                    We have parsed the second argument, and can now define macros for all permuta-
                    tions of the font attributes to point to \MT@pr@c@{name}, ...
3385 \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.
3386 \MT@gdef@n{MT@pr@c@~\MT@pr@c@name}{#3}%
3387 \MT@end@catcodes
3388 }
3389 (/pdftex-def|xetex-def|luatex-def)

\SetExpansion    \SetExpansion only differs in that it allows some extra options (stretch, shrink,
                    step, auto).
3390 (*pdftex-def|luatex-def)
3391 \def\SetExpansion{%
3392 \MT@begin@catcodes
3393 \MT@SetExpansion
3394 }

\MT@SetExpansion
  \MT@ex@c@name 3395 \newcommand*\MT@SetExpansion[3] [] {%
\MT@extra@context 3396 \let\MT@extra@context\@empty
  \MT@permutelist 3397 \MT@set@named@keys{MT@ex@c}{#1}%
3398 \MT@ifdefined@n@T{MT@ex@c@~\MT@ex@c@name @factor}{%
3399 \ifnum\csname MT@ex@c@~\MT@ex@c@name @factor\endcsname > \@m
3400 \MT@warning@n1{Expansion factor \number\@nameuse{MT@ex@c@~\MT@ex@c@name @factor}
3401 too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
3402 maximum of 1000}%
3403 \MT@glet@nc{MT@ex@c@~\MT@ex@c@name @factor}\@m
3404 \fi
3405 }%
3406 debug\MT@dinfo{1}{creating expansion list `~\MT@ex@c@name'}%
3407 \def\MT@permutelist{ex@c}%
3408 \setkeys{MT@cfg}{#2}%
3409 \MT@permute
3410 \MT@gdef@n{MT@ex@c@~\MT@ex@c@name}{#3}%
3411 \MT@end@catcodes
3412 }

\SetTracking
3413 \def\SetTracking{%
3414 \MT@begin@catcodes
3415 \MT@SetTracking
3416 }

\MT@SetTracking    Third argument may be empty.
3417 \newcommand*\MT@SetTracking[3] [] {%
3418 \let\MT@extra@context\@empty
3419 \MT@set@named@keys{MT@tr@c}{#1}%

```

```

3420 (debug)\MT@dinfo{1}{creating tracking list `\MT@tr@c@name'}%
3421 \def\MT@permutelist{tr@c}%
3422 \setkeys{MT@cfg}{#2}%
3423 \MT@permute
3424 \KV@sp@def@tempa{#3}%
3425 \MT@ifempty\tempa\relax{%
3426 \MT@ifint\tempa
3427 {\MT@xdef@n{MT@tr@c@MT@tr@c@name}{\tempa}}%
3428 {\MT@warning{Value `\tempa' is not a number in\MessageBreak
3429 tracking set `\MT@curr@set@name'}}}%
3430 \MT@end@catcodes
3431 }
3432 (/pdftex-def| luatex-def)

```

*\SetExtraSpacing*

```

3433 (*pdftex-def)
3434 \def\SetExtraSpacing{%
3435 \MT@begin@catcodes
3436 \MT@SetExtraSpacing
3437 }

```

*\MT@SetExtraSpacing*

```

\MT@sp@c@name 3438 \newcommand*\MT@SetExtraSpacing[3] [] {%
\MT@extra@context 3439 \let\MT@extra@context\empty
\MT@permutelist 3440 \MT@set@named@keys{MT@sp@c}{#1}%
3441 (debug)\MT@dinfo{1}{creating spacing list `\MT@sp@c@name'}%
3442 \def\MT@permutelist{sp@c}%
3443 \setkeys{MT@cfg}{#2}%
3444 \MT@permute
3445 \MT@gdef@n{MT@sp@c@MT@sp@c@name}{#3}%
3446 \MT@end@catcodes
3447 }

```

*\SetExtraKerning*

```

3448 \def\SetExtraKerning{%
3449 \MT@begin@catcodes
3450 \MT@SetExtraKerning
3451 }

```

*\MT@SetExtraKerning*

```

\MT@kn@c@name 3452 \newcommand*\MT@SetExtraKerning[3] [] {%
\MT@extra@context 3453 \let\MT@extra@context\empty
\MT@permutelist 3454 \MT@set@named@keys{MT@kn@c}{#1}%
3455 (debug)\MT@dinfo{1}{creating kerning list `\MT@kn@c@name'}%
3456 \def\MT@permutelist{kn@c}%
3457 \setkeys{MT@cfg}{#2}%
3458 \MT@permute
3459 \MT@gdef@n{MT@kn@c@MT@kn@c@name}{#3}%
3460 \MT@end@catcodes
3461 }
3462 (/pdftex-def)

```

*\MT@set@named@keys* We first set the name (if specified), then remove it from the list, and set the  
*\MT@options* remaining keys.

```

3463 (*package)
3464 \def\MT@set@named@keys#1#2{%
3465 \def\x##1name##2,##3\@nil{%
3466 \setkeys{#1}{name##2}%
3467 \gdef\MT@options{#1#3}%
3468 \MT@rem@from@clist{name=}\MT@options
3469 }%
3470 \x#2,name=,\@nil
3471 \@expandtwoargs\setkeys{#1}\MT@options
3472 }

```

`\MT@define@code@key` Define the keys for the configuration lists (which are setting the codes, in pdfTEX speak).

```
3473 \def\MT@define@code@key#1#2{%
3474   \define@key{MT@#2}{#1}[]{}%
3475   \@tempcnta=\@ne
3476   \MT@map@clist@{##1}{%
3477     \KV@sp@def\MT@val{###1}%
```

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

```
3478     \MT@get@highlevel{#1}%
3479     \MT@edef@{MT@temp#1\the\@tempcnta}{\MT@val}%
3480     \advance\@tempcnta \@ne
3481   }%
3482 }%
3483 }
```

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

```
3484 \def\MT@define@code@key@family#1{%
3485   \define@key{MT@#1}{family}[]{}%
3486   \@tempcnta=\@ne
3487   \MT@map@clist@{##1}{%
3488     \KV@sp@def\MT@val{###1}%
3489     \MT@get@highlevel{family}%
3490     \ifMT@fontspec
3491       \edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x
3492     \fi
3493     \MT@edef@{MT@tempfamily\the\@tempcnta}{\MT@val}%
3494     \advance\@tempcnta \@ne
3495   }%
3496 }%
3497 }
```

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

```
3498 \def\MT@define@code@key@size#1{%
3499   \define@key{MT@#1}{size}[]{}%
3500   \MT@map@clist@{##1}{%
3501     \KV@sp@def\MT@val{###1}%
3502     \expandafter\MT@get@range\MT@val--\@nil
3503     \ifx\MT@val\relax \else
3504       \MT@exp@cs\MT@xadd\MT@tempsize{%
3505         {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
3506       \fi
3507     }%
3508 }%
3509 }
```

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

```
3510 \def\MT@define@code@key@font#1{%
3511   \define@key{MT@#1}{font}[]{}%
3512   \MT@map@clist@{##1}{%
3513     \KV@sp@def\MT@val{###1}%
3514     \MT@ifstreq\MT@val*{\def\MT@val{*/**/*/*}}\relax
3515     \expandafter\MT@get@font@and@size\MT@val////\@nil
3516     \ifMT@fontspec
3517       \edef\@tempb{\expandafter\MT@scrubfeatures\@tempb()\relax}%
3518     \fi
3519     \MT@xdef@{MT@\MT@permutelist @\@tempb\MT@extra@context}%
3520     {\csname MT@\MT@permutelist @name\endcsname}%
3521     <debug>\MT@dinfo@n1{1}{initialising: use list for font \@tempb=\MT@val
3522     <debug>       \ifx\MT@extra@context\@empty\else\MessageBreak
3523     <debug>       (context: \MT@extra@context)\fi}%
3524     \MT@exp@cs\MT@xaddb
3525     {MT@\MT@permutelist @\@tempb\MT@extra@context @size}%
3526     {{{\MT@val}{\@ne}{\MT@curr@set@name}}}%
```

```

3527 }%
3528 }%
3529 }

```

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

```

3530 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\nil{%
3531 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
3532 }

3533 \MT@define@code@key{encoding}{cfg}
3534 \MT@define@code@key{family}{cfg}
3535 \MT@define@code@key{series}{cfg}
3536 \MT@define@code@key{shape}{cfg}
3537 \MT@define@code@key@size{cfg}
3538 \MT@define@code@key@font{cfg}

```

`\MT@define@opt@key`

```

3539 \def\MT@define@opt@key#1#2{%
3540 \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
3541 \MT@xdef@n{MT@#1@c@MT@curr@set@name @#2}{##1}}%
3542 }

```

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

```

3543 \newcount\MT@listname@count
3544 \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`).

```

3545 \define@key{MT@#1@c}{name}[]{%
3546 \MT@ifempty{##1}{%
3547 \MT@ifdefined@n@TF{MT@#1@c@MT@curr@file/\the\inputlineno}{%
3548 \global\advance\MT@listname@count@ne
3549 \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno
3550 (\number\MT@listname@count)}%
3551 }{%
3552 \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
3553 }%
3554 }{%
3555 \MT@edef@n{MT@#1@c@name}{##1}%
3556 \MT@ifdefined@n@T{MT@#1@c@csname MT@#1@c@name\endcsname}{%
3557 \MT@warning{Redefining \@nameuse{MT@abbr@#1} list ~\@nameuse{MT@#1@c@name}'}%
3558 }%
3559 }%
3560 \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
3561 }%
3562 \MT@define@opt@key{#1}{load}%
3563 \MT@define@opt@key{#1}{factor}%
3564 \MT@define@opt@key{#1}{preset}%
3565 \MT@define@opt@key{#1}{inputenc}%

```

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

```

3566 \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}%
3567 }
3568 (/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.

```

3569 (*pdfTeX-def|luatex-def)
3570 (pdfTeX-def)\MT@requires@pdfTeX7{
3571 \define@key{MT@ex@c}{context}[]{%
3572 \MT@ifempty{#1}\relax{%
3573 \MT@gl@et\MT@copy@font\MT@copy@font@

```

```

3574     \def\MT@extra@context{#1}%
3575     }%
3576   }
3577   \MT@addto@setup{%
3578     \define@key{MT@ex@c}{context}[]{}%
3579     \ifx\MT@copy@font\MT@copy@font@
3580       \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3581     \else
3582       \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
3583         Ignoring `context' key\on@line}%
3584       {Either move the settings inside the preamble,\MessageBreak
3585         or load the package with the `copyfonts' option.}%
3586     \fi
3587   }%
3588 }

```

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.

```

3589   \define@key{MT@pr@c}{context}[]{}%
3590   \MT@ifempty{#1}\relax{%
3591     \MT@gl@t\MT@copy@font\MT@copy@font@
3592     \def\MT@extra@context{#1}%
3593   }%
3594 }
3595 \MT@addto@setup{%
3596   \define@key{MT@pr@c}{context}[]{}%
3597   \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3598   \ifx\MT@copy@font\MT@copy@font@\else
3599     \MT@warning@nl{If protrusion contexts don't work as expected,
3600       \MessageBreak load the package with the `copyfonts' option}%
3601   \fi
3602 }%
3603 }
3604 </pdfTeX-def|luatex-def>
3605 <*pdfTeX-def>
3606 }{
3607   \define@key{MT@ex@c}{context}[]{}%
3608   \MT@error{Expansion contexts only work with pdfTeX 1.40.4\MessageBreak
3609     or later. Ignoring `context' key\on@line}%
3610   {Upgrade pdfTeX.}%
3611 }
3612 </pdfTeX-def>
3613 <*pdfTeX-def|xetex-def>
3614   \define@key{MT@pr@c}{context}[]{}%
3615   \MT@error{Protrusion contexts only work with pdfTeX
3616     <pdfTeX-def> 1.40.4\MessageBreak or later.
3617     <xetex-def> \MessageBreak or luatex.
3618     Ignoring `context' key\on@line}%
3619   <pdfTeX-def> {Upgrade pdfTeX.}%
3620   <xetex-def> {Use pdfTeX or luatex.}%
3621 }
3622 </pdfTeX-def|xetex-def>
3623 <pdfTeX-def>}

```

\MT@warn@nodim

```

3624 <*package>
3625 \def\MT@warn@nodim#1{%
3626   \MT@warning{\`@tempa' is not a dimension.\MessageBreak
3627     Ignoring it and setting values relative to\MessageBreak #1}%
3628 }

```

3629 *(/package)*

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

```
3630 (*pdfTeX-def|xetex-def|luatex-def)
3631 \define@key{MT@pr@c}{unit}[character]{%
3632   \MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
3633   \def\@tempa{#1}%
3634   \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.

```
3635     \MT@ifdimen\@tempa
3636     {\MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
3637     {\MT@warn@nodim{character widths}}%
3638   }%
3639 }
```

3640 *(/pdfTeX-def|xetex-def|luatex-def)*

Tracking may only be relative to a dimension.

```
3641 (*pdfTeX-def|luatex-def)
3642 \define@key{MT@tr@c}{unit}[1em]{%
3643   \MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
3644   \def\@tempa{#1}%
3645   \MT@ifdimen\@tempa
3646   {\MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
3647   {\MT@warn@nodim{1em}}%
3648   \MT@gdef@n{MT@tr@c@MT@curr@set@name @unit}{1em}}%
3649 }
3650 (/pdfTeX-def|luatex-def)
```

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

```
3651 (*pdfTeX-def)
3652 \MT@map@clist@n{sp,kn}{%
3653   \define@key{MT@#1@c}{unit}[space]{%
3654     \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
3655     \def\@tempa{##1}%
3656     \MT@ifstreq\@tempa{character}\relax{%
3657       \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
3658       \MT@ifstreq\@tempa{space}\relax{%
3659         \MT@ifdimen\@tempa
3660         {\MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
3661         {\MT@warn@nodim{width of space}}%
3662       }%
3663     }%
3664   }%
3665 }
3666 (/pdfTeX-def)
```

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

```
3667 (*pdfTeX-def|luatex-def)
3668 \MT@map@clist@n{stretch,shrink,step}{%
3669   \define@key{MT@ex@c}{#1}[] {%
3670     \MT@ifempty{##1}\relax{%
3671       \MT@ifint{##1}{%
```

A space terminates the number.

```
3672     \MT@gdef@n{MT@ex@c@MT@curr@set@name @#1}{##1 }%
3673   }{%
3674     \MT@warning{%
3675       Value `##1' for option `#1' is not a number.\MessageBreak
3676       Ignoring it}%
3677   }%
3678 }%
3679 }%
3680 }
```

```

3681 \define@key{MT@ex@c}{auto}[true]{%
3682   \def\@tempa{#1}%
3683   \csname if\@tempa\endcsname

   Don't use autoexpand for pdfTeX version older than 1.20.
3684 <pdfTeX-def> \MT@requirespdfTeX4{%
3685   \MT@gdefon{MT@ex@c\MT@curr@set@name @auto}{autoexpand}%
3686 <*pdfTeX-def> }{%
3687   }{%
3688   \MT@warning{pdfTeX too old for automatic font expansion}%
3689   }
3690 </pdfTeX-def>
3691 \else
3692 <pdfTeX-def> \MT@requirespdfTeX4{%
3693   \MT@glletenc{MT@ex@c\MT@curr@set@name @auto}\@empty
3694 <pdfTeX-def> } \relax
3695 \fi
3696 }

```

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

```

3697 \MT@define@opt@key{tr}{spacing}
3698 \MT@define@opt@key{tr}{outerspacing}
3699 \MT@define@opt@key{tr}{outerkerning}

```

Which ligatures should be disabled?

```

3700 \define@key{MT@tr@c}{noligatures}[]%
3701   {\MT@xdefon{MT@tr@c\MT@curr@set@name @noligatures}{#1}}
3702 \define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing=#1}}
3703 \define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning=#1}}
3704 \define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{noligatures=#1}}
3705 </pdfTeX-def|LaTeX-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.

```

3706 <*package>
3707 \renewcommand*\DeclareCharacterInheritance[1][]{%
3708   \let\MT@extra@context\@empty
3709   \let\MT@extra@inputenc\@undefined
3710   \let\MT@inh@feat\@empty
3711   \setkeys{MT@inh@}{#1}%
3712   \MT@begin@catcodes
3713   \MT@set@inh@list
3714 }

```

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

```

3715 \def\MT@set@inh@list#1#2{%
3716   \MT@ifempty\MT@inh@feat{%
3717     \MT@map@clist@c\MT@features{\MT@declare@char@inh{##1}{#1}{#2}}%
3718   }{%
3719     \MT@map@clist@c\MT@inh@feat{%
3720       \KV@sp@def\@tempa{##1}%
3721       \MT@ifempty\@tempa\relax%

```

```

3722     \MT@exp@one@n\MT@declare@char@inh
3723     {\csname MT@rbba@\@tempa\endcsname} {#1} {#2}%
3724     }%
3725     }}%
3726     }%
3727     \MT@end@catcodes
3728     }

```

The keys for the optional argument.

```

3729 \MT@map@clist@c\MT@features@long{%
3730   \define@key{MT@inh@}{#1} [] {\edef\MT@inh@feat{\MT@inh@feat#1,}}
3731 \define@key{MT@inh@}{inputenc} {\def\MT@extra@inputenc{#1}}

```

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

```

3732 \def\MT@declare@char@inh#1#2#3{%
3733   \MT@edef@n{MT@#1@inh@name}%
3734   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3735   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3736   \MT@ifdefined@c@T\MT@extra@inputenc{%
3737     \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3738   (debug)\MT@edinfo{1}{creating inheritance list ` \@nameuse{MT@#1@inh@name}' }%
3739   \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname} {#3}%
3740   \def\MT@permutelist{#1@inh}%
3741   \setkeys{MT@inh} {#2}%
3742   \MT@permute
3743   }

```

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

```

3744 \MT@define@code@key{encoding}{inh}
3745 \MT@define@code@key@family {inh}
3746 \MT@define@code@key{series} {inh}
3747 \MT@define@code@key{shape} {inh}
3748 \MT@define@code@key@size {inh}
3749 \MT@define@code@key@font {inh}

```

`\MT@inh@do` Now parse the third argument, the inheritance lists. We define the commands `\MT@inh@<name>@<slot>@`, containing the inheriting characters. They will also be translated to slot numbers here, to save some time. The following will be executed only once, namely the first time this inheritance list is encountered (in `\MT@set@<feature>@codes`).

```

3750 \def\MT@inh@do#1,{%
3751   \ifx\relax#1\@empty \else
3752     \MT@inh@split #1==\relax
3753     \expandafter\MT@inh@do
3754   \fi
3755 }

```

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

```

3756 </package>
3757 <+pdfTeX-def|xetex-def|luatex-def>
3758 \def\MT@inh@split#1=#2=#3\relax{%
3759   \def\@tempa{#1}%
3760   \ifx\@tempa\@empty \else
3761     \MT@get@slot
3762     <pdfTeX-def|luatex-def> \ifnum\MT@char > \m@ne
3763     <xetex-def> \ifx\MT@char\@empty\else
3764       \let\MT@val\MT@char
3765       \MT@map@clist@n{#2}{%
3766         \def\@tempa{##1}%
3767       \ifx\@tempa\@empty \else

```

```

3768     \MT@get@slot
3769 (pdftex-def|luatex-def) \ifnum\MT@char > \m@ne
3770 (xetex-def) \ifx\MT@char\@empty\else
3771     \MT@exp@cs\MT@xadd\MT@inh\MT@listname @\MT@val @\{\MT@char}\}%
3772     \fi
3773     \fi
3774     }%
3775 (debug)\MT@edinfo@n1{2}{children of #1 (\MT@val):
3776 (debug) \@nameuse\MT@inh\MT@listname @\MT@val @}\}%
3777     \fi
3778     \fi
3779 }
3780 (/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.

`\MT@permute@` 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.

```

3781 (*package)
3782 \def\MT@permute{%
3783   \let\MT@cnt@encoding\@ne
3784   \MT@permute@

  Undefine commands for the next round.
3785   \MT@map@tlist@n{\encoding}{family}{series}{shape}}\MT@permute@reset
3786   \MT@gl@t\MT@tempsize\@undefined
3787 }
3788 \def\MT@permute@{%
3789   \let\MT@cnt@family\@ne
3790   \MT@permute@@
3791   \MT@increment\MT@cnt@encoding
3792   \MT@ifdefined@n@T\MT@tempencoding\MT@cnt@encoding}%
3793   \MT@permute@
3794 }
3795 \def\MT@permute@@{%
3796   \let\MT@cnt@series\@ne
3797   \MT@permute@@@
3798   \MT@increment\MT@cnt@family
3799   \MT@ifdefined@n@T\MT@tempfamily\MT@cnt@family}%
3800   \MT@permute@@@
3801 }
3802 \def\MT@permute@@@{%
3803   \let\MT@cnt@shape\@ne
3804   \MT@permute@@@@
3805   \MT@increment\MT@cnt@series
3806   \MT@ifdefined@n@T\MT@tempseries\MT@cnt@series}%
3807   \MT@permute@@@@
3808 }
3809 \def\MT@permute@@@@{%
3810   \MT@permute@@@@@
3811   \MT@increment\MT@cnt@shape
3812   \MT@ifdefined@n@T\MT@tempshape\MT@cnt@shape}%
3813   \MT@permute@@@@@
3814 }

```

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

```

3815 \def\MT@permute@@@@@{%
3816   \MT@permute@define{encoding}%

```

```

3817 \ifMT@document
3818 \ifx\MT@tempencoding\@empty \else
3819 \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
3820 {\expandafter\expandafter\expandafter\@gobble}%
3821 \fi
3822 \fi
3823 \MT@permute@@@@@
3824 }

```

\MT@permute@@@@@

```

3825 \def\MT@permute@@@@@{%
3826 \MT@permute@define{family}%
3827 \MT@permute@define{series}%
3828 \MT@permute@define{shape}%
3829 \edef\@tempa{\MT@tempencoding
3830 \MT@tempfamily
3831 \MT@tempseries
3832 \MT@tempshape
3833 \MT@ifdefined@c@T\MT@tempsize *}%

```

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

```

3834 \MT@ifstreq\@tempa{////}\relax%
3835 \ifx\MT@tempencoding\@empty
3836 \MT@warning{%
3837 You have to specify an encoding for\MessageBreak
3838 \@nameuse{MT@abbr@MT@permutelist} list
3839 ~\@nameuse{MT@MT@permutelist @name}'.\MessageBreak
3840 Ignoring it}%
3841 \else
3842 \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

3843 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3844 \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3845 }%
3846 \MT@exp@cs\MT@xaddb
3847 {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3848 \MT@tempsize
3849 (debug)\MT@dinfo@n1{1}{initialising: use list for font \@tempa,\MessageBreak
3850 (debug) sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3851 (debug) @sizes\endcsname}%
3852 }{%

```

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.

```

3853 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3854 \MT@ifstreq{\csname MT@MT@permutelist @\@tempa\MT@extra@context\endcsname}%
3855 {\csname MT@MT@permutelist @\csname MT@MT@permutelist @name\endcsname @load\endcsname}%
3856 \relax}%
3857 \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3858 ~\@nameuse{MT@MT@permutelist @name}' will override list\MessageBreak
3859 ~\@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3860 for font \@tempa'}%
3861 }%
3862 }%
3863 (debug)\MT@dinfo@n1{1}{initialising: use list for font \@tempa
3864 (debug) \ifx\MT@extra@context\@empty\else\MessageBreak
3865 (debug) (context: \MT@extra@context)\fi}%
3866 }%
3867 \MT@xdefn{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3868 {\csname MT@MT@permutelist @name\endcsname}%
3869 \fi
3870 }%
3871 }

```

```

\MT@permute@define    Define the commands.
3872 \def\MT@permute@define#1{%
3873   \@tempcnta=\csname MT@cnt@#1\endcsname\relax
3874   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3875   {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3876   {\MT@let@nc{MT@temp#1}\empty}%
3877 }

\MT@permute@reset    Reset the commands.
3878 \def\MT@permute@reset#1{%
3879   \@tempcnta=\@ne
3880   \MT@loop
3881   \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3882   \advance\@tempcnta\@ne
3883   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3884   \iftrue
3885   \iffalse
3886   \MT@repeat
3887 }

\MT@check@rlist    For every new range item in \MT@tempsize, check whether it overlaps with ranges
                   in the existing list.
3888 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

\MT@check@rlist@    Define the current new range and ...
3889 \def\MT@check@rlist@#1#2#3{%
3890   \def\@tempb{#1}%
3891   \def\@tempc{#2}%
3892   \MT@iffalse
3893   \MT@exp@cs\MT@map@tlist@c
3894   {MT\MT@permutelist @\@tempa\MT@extra@context @sizes}%
3895   \MT@check@range
3896 }

\MT@check@range    ... recurse through the list of existing ranges.
3897 \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.
3898 \def\MT@check@range@#1#2#3{%
3899   \MT@ifdim{#2}=\m@ne{%
3900     \MT@ifdim\@tempc=\m@ne{%


- Both items are simple sizes.


3901     \MT@ifdim\@tempb={#1}\MT@if@true\relax
3902     }{%


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


3903     \MT@ifdim\@tempb>{#1}\relax{%
3904       \MT@ifdim\@tempc>{#1}{%
3905         \MT@if@true
3906         \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3907       }\relax
3908     }%
3909   }%
3910 }{%


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


3912   \MT@ifdim\@tempb<{#2}{%
3913     \MT@ifdim\@tempb<{#1}\relax\MT@if@true

```

```

3914     }\relax
3915     }{%

• Both items are ranges.

3916     \MT@ifdim\@tempb<{#2}{%
3917         \MT@ifdim\@tempc>{#1}{%
3918         \MT@iftrue
3919         \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3920     }\relax
3921     }\relax
3922     }%
3923 }%
3924 \ifMT@if@
3925 \MT@ifstreq{#3}%
3926     {\csname MT@MT@permutelist @\csname MT@MT@permutelist @name\endcsname @load\endcsname}%
3927     \relax}%
3928     \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3929     ~\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
3930     list ~#3' for font \@tempa,\MessageBreak size \@tempb}%
3931     }%

```

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

```

3932     \expandafter\MT@tlist@break
3933     \fi
3934 }

```

## 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 3935 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 3936 \newif\ifMT@opt@auto
3937 \newif\ifMT@opt@DVI

\MT@optwarn@admissible    Some warnings.
3938 \def\MT@optwarn@admissible#1#2{%
3939     \MT@warning@nl{`#1' is not an admissible value for option\MessageBreak
3940         ~#2'. Assuming `false'}%
3941 }

\MT@optwarn@nan
3942 </package>
3943 <*package|letterspace>
3944 <plain>\MT@requires@latex1{
3945     \def\MT@optwarn@nan#1#2{%
3946         \MT@warning@nl{Value ~#1' for option ~#2' is not a\MessageBreak number.
3947             Using default value of \number\@nameuse{MT@#2@default}}%
3948     }
3949     <plain>}\relax
3950 </package|letterspace>
3951 <*package>

\MT@opt@def@set
3952 \def\MT@opt@def@set#1{%
3953     \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
3954         \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
3955     }{%
3956         \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
3957         \MT@warning@nl{The #1 set ~\MT@val' is undeclared.\MessageBreak
3958             Using set ~\@nameuse{MT@\@tempb @setname}' instead}%
3959     }%

```

3960 }

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

```

3961 \MT@map@clist@n{protrusion,expansion}{%
3962   \define@key{MT}{#1}[true]{%
3963     \csname MT@opt@#1true\endcsname
3964     \MT@map@clist@n{##1}{%
3965       \KV@sp@def\MT@val{###1}%
3966       \MT@ifempty\MT@val\relax{%
3967         \csname MT@#1true\endcsname
3968         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3969         \MT@ifstreq\MT@val{true}\relax
3970         {%
3971           \MT@ifstreq\MT@val{false}{%
3972             \csname MT@#1false\endcsname
3973           }{%
3974             \MT@ifstreq\MT@val{compatibility}{%
3975               \MT@let@nc{MT@\@tempb @level}\@ne
3976             }{%
3977               \MT@ifstreq\MT@val{nocompatibility}{%
3978                 \MT@let@nc{MT@\@tempb @level}\tw@
3979               }{%

```

If everything failed, it should be a set name.

```

3980       \MT@opt@def@set{#1}%
3981     }%
3982   }%
3983 }%
3984 }%
3985 }%
3986 }%
3987 }%
3988 }

```

activate is a shortcut for protrusion and expansion.

```

3989 \define@key{MT}{activate}[true]{%
3990   \setkeys{MT}{protrusion={#1}}%
3991   \setkeys{MT}{expansion={#1}}%
3992 }

```

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

```

3993 \MT@map@clist@n{spacing,kerning,tracking}{%
3994   \define@key{MT}{#1}[true]{%
3995     \MT@map@clist@n{##1}{%
3996       \KV@sp@def\MT@val{###1}%
3997       \MT@ifempty\MT@val\relax{%
3998         \csname MT@#1true\endcsname
3999         \MT@ifstreq\MT@val{true}\relax
4000         {%
4001           \MT@ifstreq\MT@val{false}{%
4002             \csname MT@#1false\endcsname
4003           }{%
4004             \edef\@tempb{\csname MT@rbba@#1\endcsname}%
4005             \MT@opt@def@set{#1}%
4006           }%
4007         }%
4008       }%
4009     }%
4010   }%
4011 }

```

\MT@def@boo1@opt The true/false options: draft, final (may be inherited from the class options), auto, selected, babel, DVIoutput, defersetup, copyfonts.

```

4012 \def\MT@def@boo1@opt#1#2{%

```

```

4013 \define@key{MT}{#1}[true]{%
4014 \def\@tempa{##1}%
4015 \MT@ifstreq\@tempa{true}\relax{%
4016 \MT@ifstreq\@tempa{false}\relax{%
4017 \MT@optwarn@admissible{##1}{#1}%
4018 \def\@tempa{false}%
4019 }%
4020 }%
4021 #2%
4022 }%
4023 }

```

Boolean options that only set the switch.

```

4024 \MT@map@clist@n{draft,selected,babel}{%
4025 \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
4026 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotruer}

```

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

```

4027 </package>
4028 <*pdftex-def|luatex-def|xetex-def>
4029 <luatex-def>\MT@requires@luatex4{\let\pdfoutput\outputmode}\relax
4030 \MT@def@bool@opt{DVIoutput}{%
4031 \csname if\@tempa\endcsname
4032 <*pdftex-def|luatex-def>
4033 \ifnum\pdfoutput>\z@ \MT@opt@DVItrue \fi
4034 \pdfoutput\z@
4035 \else
4036 \ifnum\pdfoutput<\@ne \MT@opt@DVItrue \fi
4037 \pdfoutput\@ne
4038 </pdftex-def|luatex-def>
4039 <xetex-def> \MT@warning@n{Ignoring `DVIoutput' option}%
4040 \fi
4041 }
4042 </pdftex-def|luatex-def|xetex-def>

```

Setting the `deferssetup` 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.

```

4043 <*package>
4044 \MT@def@bool@opt{deferssetup}{%
4045 \csname if\@tempa\endcsname \else
4046 \AtEndOfPackage{%
4047 \MT@setup@
4048 \let\MT@setup@\empty
4049 \let\MT@addto@setup\@firstofone
4050 }%
4051 \fi
4052 }
4053 </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 Lua<sub>T</sub><sub>E</sub>X 0.30 or newer.

```

4054 <*pdftex-def|luatex-def>
4055 <pdftex-def>\MT@requires@pdftex7{
4056 \MT@def@bool@opt{copyfonts}{%
4057 \csname if\@tempa\endcsname

```

```

4058     \MT@gllet\MT@copy@font\MT@copy@font@
4059     \else
4060     \MT@gllet\MT@copy@font\relax
4061     \fi
4062   }
4063   <pdfTeX-def>}{
4064   </pdfTeX-def|LaTeX-def>
4065   <*pdfTeX-def|xetex-def>
4066   \MT@def@bool@opt{copyfonts}{%
4067     \csname if\@tempa\endcsname
4068     \MT@error
4069   <pdfTeX-def>      {The pdfTeX version you are using is too old\MessageBreak
4070   <pdfTeX-def>      to use the `copyfonts' option}{Upgrade pdfTeX.}%
4071   <xetex-def>       {The `copyfonts' option does not work with xetex}
4072   <xetex-def>       {Use pdfTeX or LaTeX instead.}%
4073   \fi
4074   }
4075   <pdfTeX-def>}
4076   </pdfTeX-def|xetex-def>

```

`final` is the opposite to `draft`.

```

4077 <*package>
4078 \MT@def@bool@opt{final}{%
4079   \csname if\@tempa\endcsname
4080   \MT@draftfalse
4081   \else
4082   \MT@drafttrue
4083   \fi
4084 }

```

For verbose output, we redefine `\MT@vinfo`.

```

4085 \define@key{MT}{verbose}[true]{%
4086   \let\MT@vinfo\MT@info@n1
4087   \def\@tempa{#1}%
4088   \MT@ifstreq\@tempa{true}\relax%

```

Take problems seriously.

```

4089   \MT@ifstreq\@tempa{errors}{%
4090     \let\MT@warning \MT@warn@err
4091     \let\MT@warning@n1\MT@warn@err
4092   }{%
4093     \let\MT@vinfo\@gobble

```

Cast warnings to the winds.

```

4094   \MT@ifstreq\@tempa{silent}{%
4095     \let\MT@warning \MT@info
4096     \let\MT@warning@n1\MT@info@n1
4097   }{%
4098     \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
4099   }%
4100 }%
4101 }%
4102 }
4103 </package>

```

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

```

4104 <*package|letterspace>
4105 <plain>\MT@requires@l@tex1{
4106 \MT@map@clist@n{%
4107 <package> stretch,shrink,step,%
4108 letterspace}{%
4109 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
4110 \def\@tempa{##1 }%

```

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

```

4111 \MT@ifint\@tempa
4112   {\MT@edef@n{MT@#1}{\@tempa}}%
4113   {\MT@optwarn@nan{##1}{#1}}%
4114 }%
4115 }
4116 <plain>\relax
4117 </package|letterspace>

```

factor will define the protrusion factor only.

```

4118 <*package>
4119 \define@key{MT}{factor}[\MT@factor@default]{%
4120   \def\@tempa{#1 }%
4121   \MT@ifint\@tempa
4122   {\edef\MT@pr@factor{\@tempa}}
4123   {\MT@optwarn@nan{#1}{factor}}%
4124 }

```

Unit for protrusion codes.

```

4125 \define@key{MT}{unit}[character]{%
4126   \def\@tempa{#1}%
4127   \MT@ifstreq\@tempa{character}\relax{%
4128     \MT@ifdimen\@tempa
4129     {\let\MT@pr@unit\@tempa}%
4130     {\MT@warning@n1{\@tempa' is not a dimension.\MessageBreak
4131       Ignoring it and setting values relative to\MessageBreak
4132       character widths}}%
4133   }%
4134 }

```

#### 14.4.2 Loading the definition file

`\MT@endinput` Abort if no capable engine found.

```

4135 \let\MT@endinput\relax
4136 \ifx\MT@engine\relax
4137   \MT@warning@n1{You don't seem to be using pdftex, luatex or xetex.\MessageBreak
4138     \MT@MT' only works with these engines.\MessageBreak
4139     I will quit now}
4140   \MT@clear@options
4141 \else

```

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

```

4142   \input{microtype-\MT@engine tex.def}
4143 \fi
4144 \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.)

```

4145 \MT@protrusiontrue
4146 </package>
4147 <*pdftex-def|luatex-def>
4148 \ifnum\pdfoutput<\@ne \else

```

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

```

4149 <pdftex-def> \MT@requires@pdftex4{

```

```

4150 \MT@expansiontrue
4151 \MT@autottrue
4152 (pdftex-def) }\relax
4153 \fi
4154 (pdftex-def|luatex-def)

```

The main configuration file will be loaded before processing the package options.

`\MT@config@file` However, the config option must of course be evaluated beforehand. We also have  
`\MT@get@config` to define a no-op for the regular option processing later.

```

4155 (*package)
4156 \define@key{MT}{config}[]{\relax}
4157 \def\MT@get@config#1config=#2,#3\@nil{%
4158 \MT@ifempty{#2}%
4159 {\def\MT@config@file{\MT@MT.cfg}}%
4160 {\def\MT@config@file{#2.cfg}}%
4161 }
4162 \expandafter\expandafter\expandafter\MT@get@config
4163 \csname opt@\@currname.\@currxt\endcsname,config=,\@nil

```

Load the file.

```

4164 \IfFileExists{\MT@config@file}{%
4165 \MT@info@nl{Loading configuration file \MT@config@file}%
4166 \MT@begin@catcodes
4167 \let\MT@begin@catcodes\relax
4168 \let\MT@end@catcodes\relax
4169 \let\MT@curr@file\MT@config@file
4170 \input{\MT@config@file}%
4171 \endgroup
4172 }{\MT@warning@nl{%
4173 Could not find configuration file \MT@config@file!\MessageBreak
4174 This will almost certainly cause undesired results.\MessageBreak
4175 Please fix your installation}%
4176 }

```

`\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).

```

4177 \def\MT@check@active@set#1{%
4178 \MT@ifdefined@n@TF{MT@#1@setname}{%
4179 \MT@info@nl{Using \nameuse{MT@abbr@#1} set \nameuse{MT@#1@setname}'}%
4180 }{%
4181 \MT@ifdefined@n@TF{MT@default@#1@set}{%
4182 \MT@gl@et@nn{MT@#1@setname}{MT@default@#1@set}%
4183 \MT@info@nl{Using default \nameuse{MT@abbr@#1} set \nameuse{MT@#1@setname}'}%
4184 }{%

```

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

```

4185 \MT@gdef@n{MT@#1@setname}{@}%
4186 \MT@warning@nl{No \nameuse{MT@abbr@#1} set chosen, no default set declared.
4187 \MessageBreak Using empty set}%
4188 }%
4189 }%
4190 }

```

#### 14.4.4 Hook for other packages

`\Microtype@Hook` This hook may be used by font package authors, e.g., to declare alias fonts. If it is defined, it will be executed here, i.e., after the main configuration file has been loaded, and before the package options are evaluated.

This hook was needed in versions prior to 1.9a to overcome the situation that

(1) the microtype package should be loaded after all font defaults have been set up (hence, using `\ifpackageloaded` in the font package was not viable), and (2) checking `\AtBeginDocument` could be too late, since fonts might already have been loaded, and consequently set up, in the preamble. With the new deferred setup, one could live without this command, however, it remains here since it's simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

Package authors should check whether the command is already defined so that existing definitions by other packages aren't overwritten. Example:

```
\def\MinionPro@MT@Hook{\DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}}
\ifpackageloaded{microtype}
  \MinionPro@MT@Hook
  {\ifundefined{Microtype@Hook}
   {\let\Microtype@Hook\MinionPro@MT@Hook}
   {\g@addto@macro\Microtype@Hook{\MinionPro@MT@Hook}}}
```

`\MicroType@Hook` with a capital T (which only existed in version 1.7) is provided for compatibility reasons. At some point in the future, it will no longer be available, hence it should not be used.

```
4191 \MT@ifdefined@ct\MicroType@Hook{\MT@warning{%
4192   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
4193   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
4194 \MT@ifdefined@ct\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.

```
4195 \def\microtypesetup{\setkeys{MT}}
4196 \MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
4197 /package
4198 <*pdfTeX-def|luatex-def|xetex-def
4199 \def\MT@define@optionX#1#2{%
4200   \define@key{MTX}{#1}[true]{%
4201     \edef@tempb{\csname MT@rbba@#1\endcsname}%
4202     \MT@map@clist@n{##1}{%
4203       \KV@esp@def\MT@val{###1}%
4204       \MT@ifempty\MT@val\relax{%
4205         \@tempcnta=\m@ne
4206         \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.

```
4207   \MT@checksetup{#1}{%
4208     \@tempcnta=\csname MT@@tempb @level\endcsname
4209     \MT@vinfo{Enabling #1
4210       (level \number\csname MT@@tempb @level\endcsname)\online}%
4211   }%
4212 }{%
4213   \MT@ifstreq\MT@val{false}{%
4214     \@tempcnta=\z@
4215     \MT@vinfo{Disabling #1\online}%
4216   }{%
4217     \MT@ifstreq\MT@val{compatibility}{%
4218       \MT@checksetup{#1}{%

```

```

4219         \@tempcnta=\@ne
4220         \MT@let@nc{MT@\@tempb @level}\@ne
4221         \MT@vinfo{Setting #1 to level 1\on@line}%
4222     }%
4223 }{%
4224     \MT@ifstreq\MT@val{nocompatibility}{%
4225         \MT@checksetup{#1}{%
4226             \@tempcnta=\tw@
4227             \MT@let@nc{MT@\@tempb @level}\tw@
4228             \MT@vinfo{Setting #1 to level 2\on@line}%
4229         }%
4230     }{\MT@error{Value `MT@val' for key `#1' not recognised}
4231         {Use any of `true', `false', `compatibility' or
4232         `nocompatibility'.}%
4233     }%
4234 }%
4235 }%
4236 }%
4237 \ifnum\@tempcnta>\m@ne
4238 #2\@tempcnta\relax
4239 \fi
4240 }%
4241 }%
4242 }%
4243 }

```

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

```

4244 \def\MT@checksetup#1{%
4245     \csname ifMT@#1\endcsname
4246     \expandafter\@firstofone
4247     \else
4248         \MT@error{You cannot enable #1 if it was disabled\MessageBreak
4249             in the package options}{Load microtype with #1 enabled.}%
4250     \expandafter\@gobble
4251     \fi
4252 }

4253 \MT@define@optionX{protrusion}\MT@protrudechars
4254 /pdftex-def|luatex-def|xetex-def
4255 *pdftex-def|luatex-def
4256 \MT@define@optionX{expansion}\MT@adjustspacing

```

`\MT@protrudechars`

```

\MT@adjustspacing 4257 *luatex-def
4258 \MT@requires@luatex4{
4259     \let\pdfprotrudechars\protrudechars
4260     \let\pdfadjustspacing\adjustspacing
4261 }\relax
4262 /luatex-def
4263 \let\MT@protrudechars\pdfprotrudechars
4264 \let\MT@adjustspacing\pdfadjustspacing
4265 /pdftex-def|luatex-def
4266 *xetex-def
4267 \let\MT@protrudechars\XeTeXprotrudechars
4268 \define@key{MTX}{expansion}[true]{\MT@warning{Ignoring expansion setup}}
4269 /xetex-def

```

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

```

4270 *pdftex-def|luatex-def
4271 (pdftex-def)\MT@requires@pdftex6{
4272 (luatex-def)\MT@requires@luatex3{
4273     \def\MT@define@optionX@#1#2{%
4274         \define@key{MTX}{#1}[true]{%
4275             \MT@map@clist@n{##1}{%

```

```

4276     \KV@sp@def\MT@val{###\#1}%
4277     \MT@ifempty\MT@val\relax{%
4278       \@tempcnta=\m@ne
4279       \MT@ifstreq\MT@val{true}{%
4280         \MT@checksetup{#1}{%
4281           \@tempcnta=\@ne
4282           \MT@vinfo{Enabling #1\on@line}%
4283         }%
4284       }{%
4285         \MT@ifstreq\MT@val{false}{%
4286           \@tempcnta=\z@
4287           \MT@vinfo{Disabling #1\on@line}%
4288         }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
4289           {Use either `true' or `false'}}%
4290       }%
4291     }%
4292     \ifnum\@tempcnta>\m@ne
4293       #2\relax
4294     \fi
4295   }%
4296 }%
4297 }%
4298 }

```

We cannot simply let `\MT@tracking relax`, since this may select the already letter-spaced font instance.

```

4299   \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
4300     \else \let\MT@tracking\MT@tracking@ \fi}
4301 (pdftex-def) \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
4302 (pdftex-def) \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
4303 (pdftex-def) \pdfappendkern\@tempcnta}
4304 }{
4305 (/pdftex-def|luatex-def)
4306 (*pdftex-def|luatex-def|xetex-def)

```

Disable for older pdf<sub>T</sub>E<sub>X</sub> versions and for X<sub>Y</sub>T<sub>E</sub>X and Lua<sub>T</sub>E<sub>X</sub>.

```

4307 \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
4308 (luatex-def)
4309 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
4310 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
4311 (pdftex-def)
4312 \define@key{MTX}{activate}[true]{%
4313   \setkeys{MTX}{protrusion={#1}}}%
4314 (pdftex-def|luatex-def) \setkeys{MTX}{expansion={#1}}%
4315 }
4316 (/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.

```

4317 (*package)
4318 \let\MT@saved@setupfont\MT@setupfont
4319 \define@key{MTX}{disable}[]{%
4320   \MT@info{Inactivate `~\MT@MT' package}%
4321   \let\MT@setupfont\relax
4322 }
4323 \define@key{MTX}{enable}[]{%
4324   \MT@info{Reactivate `~\MT@MT' package}%
4325   \let\MT@setupfont\MT@saved@setupfont
4326 }
4327 (/package)

```

### 14.4.6 Processing the options

```

\MT@ProcessOptionsWithKV    Parse options.
4328 <*package|letterspace>
4329 <plain>\MT@requires@latex1{
4330 \def\MT@ProcessOptionsWithKV#1{%
4331   \let\@tempc\relax
4332   \let\MT@temp\@empty
4333 <plain> \MT@requires@latex2{
4334   \MT@map@clist@c\@classoptionslist{%
4335     \def\CurrentOption{##1}%
4336     \MT@ifdefined@nT{KV@#1@}\expandafter\MT@getkey\CurrentOption=\@nil}{%
4337       \edef\MT@temp{\MT@temp,\CurrentOption,}%
4338       \expandtwoargs\@removeelement\CurrentOption
4339         \@unusdooptionlist\@unusdooptionlist
4340     }%
4341   }%
4342   \edef\MT@temp{\noexpand\setkeys{#1}%
4343     {\MT@temp\optionlist{\@currname.\@current}}}%

plain can handle package options.
4344 <*plain>
4345 }{\edef\MT@temp{\noexpand\setkeys{#1}%
4346   {\csname usepkg@options@usepkg@pkg\endcsname}}
4347 </plain>
4348 \MT@temp
4349 \MT@clear@options
4350 }

```

\MT@getkey For key=val in class options.

```

4351 \def\MT@getkey#1=#2@nil{#1}
4352 \MT@ProcessOptionsWithKV{MT}
4353 <plain>\relax
4354 </package|letterspace>
4355 <*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).

```

4356 \MT@addto@setup{%
4357 \ifMT@draft

```

We disable most of what we've just defined in the 4357 lines above if we are running in draft mode.

```

4358 \MT@warning@n1{'draft' option active.\MessageBreak
4359   Disabling all micro-typographic extensions.\MessageBreak
4360   This might lead to different line and page breaks}%
4361 \let\MT@setupfont\relax
4362 \renewcommand*{LoadMicrotypeFile[1]}{}%
4363 \renewcommand*\microtypesetup[1]{}%
4364 \renewcommand*\microtypecontext[1]{}%
4365 \renewcommand*\lsstyle{}%
4366 \else
4367 \MT@setup@PDF
4368 \MT@setup@copies

```

Fix the font sets.

```

4369 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
4370 \MT@setup@protrusion
4371 \MT@setup@expansion
4372 \MT@setup@tracking
4373 \MT@setup@warntracking
4374 \MT@setup@spacing
4375 \MT@setup@kerning

```

```
4376 \MT@setup@noligatures
4377 }
4378 </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!

```
4379 <*pdftex-def|luatex-def>
4380 \def\MT@setup@PDF{%
4381 \MT@info@n1{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
4382 \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
4383 }
```

\MT@setup@copies Working on font copies?

```
4384 \def\MT@setup@copies{%
4385 \ifx\MT@copy@font\relax\else \MT@info@n1{Using font copies for contexts}\fi
4386 }
4387 </pdftex-def|luatex-def>
4388 <*xetex-def>
4389 \let\MT@setup@PDF\relax
4390 \let\MT@setup@copies\relax
4391 </xetex-def>
```

\MT@setup@protrusion Protrusion.

```
4392 <*pdftex-def|xetex-def|luatex-def>
4393 \def\MT@setup@protrusion{%
4394 \ifMT@protrusion
4395 \edef\MT@active@features{\MT@active@features,pr}%
4396 \MT@protrudechars\MT@pr@level
4397 \MT@info@n1{Character protrusion enabled (level \number\MT@pr@level)%
4398 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
4399 factor: \number\MT@pr@factor\fi
4400 \ifx\MT@pr@unit@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
4401 \MT@check@active@set{pr}%
4402 \else
4403 \let\MT@protrusion\relax
4404 \MT@info@n1{No character protrusion}%
4405 \fi
4406 }
4407 </pdftex-def|xetex-def|luatex-def>
```

\MT@setup@expansion For DVI output, the user must have explicitly passed the expansion option to the package.

```
4408 <*pdftex-def|luatex-def>
4409 \def\MT@setup@expansion{%
4410 \ifnum\pdfoutput<\@ne
4411 \ifMT@opt@expansion \else
4412 \MT@expansionfalse
4413 \fi
4414 \fi
4415 \ifMT@expansion
```

Set up the values for font expansion: if stretch has not been specified, we take the default value of 20.

```
4416 \ifnum\MT@stretch=\m@ne
4417 \let\MT@stretch\MT@stretch@default
4418 \fi
```

If `shrink` has not been specified, it will inherit the value from `stretch`.

```
4419 \ifnum\MT@shrink=\m@ne
4420 \let\MT@shrink\MT@stretch
4421 \fi
```

If `step` has not been specified, we will just set it to 1 for recent pdfTeX versions. My tests did not show much difference neither in compilation time (within the margin of error) nor in file size (less than 1% difference for `microtype.pdf` with `step=1` compared to `step=5`). With older versions, we set it to  $\min(\text{stretch}, \text{shrink})/5$ , rounded off, minimum value 1.

```
4422 \ifnum\MT@step=\m@ne
4423 (pdfTeX-def) \MT@requires@pdfTeX6{%
4424 \def\MT@step{1 }%
4425 (*pdfTeX-def)
4426 }{%
4427 \ifnum\MT@stretch>\MT@shrink
4428 \ifnum\MT@shrink=\z@
4429 \@tempcnta=\MT@stretch
4430 \else
4431 \@tempcnta=\MT@shrink
4432 \fi
4433 \else
4434 \ifnum\MT@stretch=\z@
4435 \@tempcnta=\MT@shrink
4436 \else
4437 \@tempcnta=\MT@stretch
4438 \fi
4439 \fi
4440 \divide\@tempcnta 5\relax
4441 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
4442 \edef\MT@step{\number\@tempcnta\space}%
4443 }%
4444 (/pdfTeX-def)
4445 \fi
4446 \ifnum\MT@step=\z@
4447 \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
4448 Setting it to one}%
4449 \def\MT@step{1 }%
4450 \fi
```

`\MT@auto` Automatic expansion of the font? This new feature of pdfTeX 1.20 makes the `fix` programme really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdfTeX).

```
4451 \let\MT@auto\@empty
4452 \ifMT@auto
```

We turn off automatic expansion if output mode is DVI and we’re running pdfTeX.

```
4453 (*pdfTeX-def)
4454 \MT@requires@pdfTeX4{%
4455 \ifnum\pdfoutput<\@ne
4456 \ifMT@opt@auto
4457 \MT@error{%
4458 Automatic font expansion only works for PDF output.\MessageBreak
4459 However, you are creating a DVI file}
4460 {If you have created expanded fonts instances, remove ‘auto’ from%
4461 \MessageBreak the package options. Otherwise, you have to switch
4462 off expansion.\MessageBreak completely.}%
4463 \fi
4464 \MT@autofalse
4465 \else
4466 (/pdfTeX-def)
4467 \def\MT@auto{autoexpand}%
4468 (*pdfTeX-def)
```

```

4469     \fi
Also, if pdfTeX is too old.
4470     }{%
4471     \MT@error{%
4472     The pdftex version you are using is too old for\MessageBreak
4473     automatic font expansion}%
4474     {If you have created expanded fonts instances, remove `auto' from\MessageBreak
4475     the package options. Otherwise, you have to switch off expansion\MessageBreak
4476     completely, or upgrade pdftex to version 1.20 or newer.}%
4477     \MT@autofalse
4478     \def\MT@auto{1000 }%
4479     }%
4480 \else

```

No automatic expansion.

```

4481     \MT@requires@pdftex4\relax{%
4482     \def\MT@auto{1000 }%
4483     }%
4484 (/pdfTeX-def)
4485     \fi

```

Choose the appropriate macro for selected expansion.

```

4486     \ifMT@selected
4487     \let\MT@set@ex@codes\MT@set@ex@codes@s
4488     \else
4489     \let\MT@set@ex@codes\MT@set@ex@codes@n
4490     \fi

```

Filter out stretch=0, shrink=0, since it would result in a pdfTeX error.

```

4491     \ifnum\MT@stretch=\z@
4492     \ifnum\MT@shrink=\z@
4493     \MT@warning@nl{%
4494     Both the stretch and shrink limit are set to zero.\MessageBreak
4495     Disabling font expansion}%
4496     \MT@expansionfalse
4497     \fi
4498     \fi
4499     \fi
4500     \ifMT@expansion
4501     \edef\MT@active@features{\MT@active@features,ex}%
4502     \MT@adjustspacing\MT@ex@level
4503     \MT@info@nl{\ifMT@auto A\else Non-afii utomatic font expansion enabled
4504     (level \number\MT@ex@level),\MessageBreak
4505     stretch: \number\MT@stretch, shrink: \number\MT@shrink,
4506     step: \number\MT@step, \ifMT@selected\else non-\fi selected}%

```

`\MT@check@step` Check whether stretch and shrink are multiples of step.

```

4507     \def\MT@check@step##1{%
4508     \@tempcnta=\csname MT@##1\endcsname
4509     \divide\@tempcnta \MT@step
4510     \multiply\@tempcnta \MT@step
4511     \ifnum\@tempcnta=\csname MT@##1\endcsname\else
4512     \MT@warning@nl{The ##1 amount is not a multiple of step.\MessageBreak
4513     The effective maximum ##1 is \the\@tempcnta\space
4514     (step \number\MT@step)}%
4515     \fi
4516     }%
4517     \MT@check@step{stretch}%
4518     \MT@check@step{shrink}%
4519     \MT@check@active@set{ex}%

```

Inside `\showhypens`, font expansion should be disabled. (Since 2017/01/10, the  $\LaTeX$  format contains a different version for  $X_{\text{pdf}}\TeX$ , but since expansion doesn't

work with X<sub>Y</sub>T<sub>E</sub>X, we don't have to bother.)

```
4520 \CheckCommand* \showhyphens[1]{\setbox0\vbox{%
4521 \color@begingroup\everypar{ }\parfillskip\z@skip
4522 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
4523 \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}%
```

`\showhyphens` I wonder why it's defined globally (in `ltfssbas.dtx`)?

```
4524 \gdef\showhyphens##1{\setbox0\vbox{%
4525 \color@begingroup\pdfadjustspacing\z@\everypar{ }\parfillskip\z@skip
4526 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
4527 \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}%

4528 \else
4529 \let\MT@expansion\relax
4530 \MT@info@nl{No font expansion}%
4531 \fi
4532 }
4533 </pdftex-def|luatex-def>
4534 <*_xetex-def>
4535 \def\MT@setup@expansion{%
4536 \ifMT@expansion
4537 \ifMT@opt@expansion
4538 \MT@error{Font expansion does not work with xetex}
4539 {Use pdftex or luatex instead.}%
4540 \fi
4541 \fi
4542 }
4543 </xetex-def>
```

`\MT@setup@tracking` Tracking, spacing and kerning.

```
4544 <*_pdftex-def|luatex-def>
4545 <pdftex-def>\MT@requires@pdftex6{%
4546 <luatex-def>\MT@requires@luatex3{%
4547 \def\MT@setup@tracking{%
4548 \ifMT@tracking
4549 \edef\MT@active@features{\MT@active@features,tr}%
4550 \MT@info@nl{Tracking enabled}%
4551 \MT@check@active@set{tr}%
```

Enable protrusion for compensation at the line edges.

```
4552 \ifMT@protrusion\else\MT@protrudechars\@ne\fi
4553 \else
4554 \let\MT@tracking\relax
4555 \MT@info@nl{No adjustment of tracking}%
4556 \fi
4557 }
4558 </pdftex-def|luatex-def>
```

`\MT@setup@spacing`

```
4559 <*_pdftex-def>
4560 \def\MT@setup@spacing{%
4561 \ifMT@spacing
4562 \edef\MT@active@features{\MT@active@features,sp}%
4563 \pdfadjustinterwordglue\@ne
4564 \MT@info@nl{Adjustment of interword spacing enabled}%
```

The `ragged2e` package sets interword spaces to a fixed value without glue. `microtye`'s modifications can therefore have undesired effects. Therefore, we issue a warning.

```
4565 \MT@with@package@T{ragged2e}{%
4566 \MT@warning@nl{You are using the `ragged2e' package.\MessageBreak
4567 Adjustment of interword spacing may lead to\MessageBreak
4568 undesired results when used with `ragged2e'.\MessageBreak
4569 In this case, disable the `spacing' option}%
4570 }%
4571 \MT@check@active@set{sp}%
```



```

4624 <pdf $\textit{tex}$ -def>          {Upgrade pdf $\textit{tex}$ .}%
4625 <luat $\textit{ex}$ -def>|xet $\textit{ex}$ -def>    {Use pdf $\textit{tex}$  instead.}%
4626     \csname MT@#1false\endcsname
4627     \MT@let@nc{MT@#1}\relax
4628     \else
4629     \MT@info@n1{No adjustment of #1%
4630 <pdf $\textit{tex}$ -def>          \space(pdf $\textit{tex}$  too old)%
4631     }%
4632     \fi
4633   }
4634 <pdf $\textit{tex}$ -def>|xet $\textit{ex}$ -def> \def\MT@setup@tracking{\MT@error@doesnt@work{tracking}}
4635   \def\MT@setup@kerning {\MT@error@doesnt@work{kerning}}
4636   \def\MT@setup@spacing {\MT@error@doesnt@work{spacing}}
4637 <pdf $\textit{tex}$ -def>})
4638 </pdf $\textit{tex}$ -def>|xet $\textit{ex}$ -def>|luat $\textit{ex}$ -def>

```

\MT@setup@warntracking

```

4639 <letterspace>\MT@addto@setup
4640 <pdf $\textit{tex}$ -def>|luat $\textit{ex}$ -def>\def\MT@setup@warntracking

```

\MT@warn@tracking@DVI

With pdf $\textit{T}\textit{E}\textit{X}$ , 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.

```

4641 <*pdf $\textit{tex}$ -def>|luat $\textit{ex}$ -def>|letterspace>
4642 {%
4643 <*pdf $\textit{tex}$ -def>|letterspace>
4644   \ifnum\pdfoutput<\@ne
4645   \def\MT@warn@tracking@DVI{%
4646 <letterspace>          \MT@pdf@or@lua%
4647     \MT@warning@n1{%
4648       You are using tracking/letterspacing in DVI mode.\MessageBreak
4649       This will probably not work, unless the post-\MessageBreak
4650       processing program (dvips, dvi $\textit{p}$ dfm(x), ...) is\MessageBreak
4651       able to create the virtual fonts on the fly}%
4652 <letterspace>          }\relax
4653     \MT@glet\MT@warn@tracking@DVI\relax
4654   }%
4655   \else
4656 </pdf $\textit{tex}$ -def>|letterspace>
4657   \def\MT@warn@tracking@DVI{%
4658     \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
4659     \MT@glet\MT@warn@tracking@DVI\relax
4660   }%
4661 <pdf $\textit{tex}$ -def>|letterspace> \fi
4662   \ifnum\MT@letterspace=\m@ne
4663   \let\MT@letterspace\MT@letterspace@default
4664   \else
4665     \MT@ls@too@large\MT@letterspace
4666   \fi
4667 }
4668 </pdf $\textit{tex}$ -def>|luat $\textit{ex}$ -def>|letterspace>
4669 <xet $\textit{ex}$ -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.

```

4670 <*pdf $\textit{tex}$ -def>|luat $\textit{ex}$ -def>
4671 \def\MT@setup@noligatures{%
4672 <pdf $\textit{tex}$ -def> \MT@requires@pdf $\textit{tex}$ 5{%
4673   \ifMT@noligatures \else
4674   \let\MT@noligatures\relax
4675   \fi
4676 <pdf $\textit{tex}$ -def> }\relax
4677 }
4678 </pdf $\textit{tex}$ -def>|luat $\textit{ex}$ -def>

```

```
4679 <xetex-def>\let\MT@setup@noligatures\relax
```

Remove the leading comma in `\MT@active@features`, and set the document switch to true.

```
4680 <*package>
4681 \MT@addto@setup{%
4682   \ifx\MT@active@features\empty \else
4683     \edef\MT@active@features{\expandafter@gobble\MT@active@features}%
4684   \fi
4685   \MT@documenttrue
4686 }
```

`\MT@set@babel@context`    Interaction with babel.

```
4687 \def\MT@set@babel@context#1{%
4688   \MT@ifdefined@n@TF{MT@babel@#1}{%
4689     \MT@info{*** Changing to language context `#1'\MessageBreak\on@line}%
4690     \expandafter\MT@exp@one@n\expandafter\microtypecontext
4691     \csname MT@babel@#1\endcsname
4692   }{%
4693     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
4694   }%
4695 }
```

`\MT@shorthandoff`    Active characters can only be switched off if babel isn't loaded after microtype.

```
4696 \@ifpackageloaded{babel}{
4697   \def\MT@shorthandoff#1#2{%
4698     \MT@info@n1{Switching off #1 babel's active characters (#2)}%
4699     \shorthandoff{#2}}
4700 }{
4701   \def\MT@shorthandoff#1#2{%
4702     \MT@error{You must load `babel' before `~\MT@MT'}
4703     {Otherwise, `~\MT@MT' cannot switch off #1 babel's\MessageBreak
4704     active characters.}}
4705 }
```

We patch the language switching commands to enable language-dependent setup.

```
4706 \MT@addto@setup{%
4707   \ifMT@babel
4708     \@ifpackageloaded{babel}{%
4709       \MT@info@n1{Redefining babel's language switching commands}%
4710       \let\MT@orig@select@language\select@language
4711       \def\select@language#1{%
4712         \MT@orig@select@language{#1}%
4713         \MT@set@babel@context{#1}%
4714       }%
4715       \let\MT@orig@foreign@language\foreign@language
4716       \def\foreign@language#1{%
4717         \MT@orig@foreign@language{#1}%
4718         \MT@set@babel@context{#1}%
4719       }%
4720     \ifMT@kerning
```

Disable French babel's active characters.

```
4721     \MT@if@false
4722     \MT@with@babel@and@T{french} \MT@if@true
4723     \MT@with@babel@and@T{frenchb} \MT@if@true
4724     \MT@with@babel@and@T{français}\MT@if@true
4725     \MT@with@babel@and@T{canadien}\MT@if@true
4726     \MT@with@babel@and@T{acadian} \MT@if@true
4727     \ifMT@if@~\MT@shorthandoff{French}{:;!?)\fi
```

Disable Turkish babel's active characters.

```
4728     \MT@if@false
4729     \MT@with@babel@and@T{turkish} \MT@if@true
4730     \ifMT@if@~\MT@shorthandoff{Turkish}{:!=)\fi
```

4731       \fi

In case babel was loaded before microtype:

```
4732       \MT@set@babel@context\languagename
4733       }{%
4734       \MT@warning@n1{You did not load the babel package.\MessageBreak
4735       The `babel' option won't have any effect}%
4736       }%
4737       \fi
4738 }
```

Now we close the \fi from \ifMT@draft.

```
4739 \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.

```
4740       \selectfont}
```

\MT@curr@file    This is the current file (hopefully with the correct extension).

```
4741 \edef\MT@curr@file{\jobname.tex}
4742 </package>
```

Finally, execute the setup macro at the end of the preamble, and empty it (the combine class calls it repeatedly).

```
4743 <*package|letterspace>
4744 <plain>\MT@requires@l@texl{
4745 \AtBeginDocument{\MT@setup@ \MT@gl@et\MT@setup@ \@empty}
4746 <plain>}\relax
4747 </package|letterspace>
```

Must come at the very, very end.

```
4748 <package>\MT@ifdefined@c@T\MT@setup@spacing@check
4749 <package> {\AtBeginDocument{\MT@setup@spacing@check}}
```

Restore catcodes.

```
4750 <package|letterspace>\MT@restore@catcodes
```

That was that.

## 15 Configuration files

Let's now write the font configuration files.

```
4751 (*config)
4752
```

### 15.1 Font sets

We first declare some sets in the main configuration file.

```
4753 (*m-t)
4754 %%% -----
4755 %%% FONT SETS
4756
4757 \DeclareMicrotypeSet{all}
4758   { }
4759
4760 \DeclareMicrotypeSet{allmath}
4761   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U} }
4762
4763 \DeclareMicrotypeSet{alltext}
4764   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU} }
4765
4766 \DeclareMicrotypeSet{allmath-nott}
4767   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U},
4768     family   = {rm*,sf*}
4769   }
4770
4771 \DeclareMicrotypeSet{alltext-nott}
4772   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
4773     family   = {rm*,sf*}
4774   }
4775
4776 \DeclareMicrotypeSet{basicmath}
4777   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,OML,OMS},
4778     family   = {rm*,sf*},
4779     series   = {md*},
4780     size     = {normalsize,footnotesize,small,large}
4781   }
4782
4783 \DeclareMicrotypeSet{basictext}
4784   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU},
4785     family   = {rm*,sf*},
4786     series   = {md*},
4787     size     = {normalsize,footnotesize,small,large}
4788   }
4789
4790 \DeclareMicrotypeSet{smallcaps}
4791   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
4792     shape    = {sc*,si,scit}
4793   }
4794
4795 \DeclareMicrotypeSet{footnotesize}
4796   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
4797     size     = {-small}
4798   }
4799
4800 \DeclareMicrotypeSet{scriptsize}
4801   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
```

```

4802     size      = {-footnotesize}
4803   }
4804
4805 \DeclareMicrotypeSet{normal font}
4806   { font = */*/*/*/* }
4807

```

The default sets.

```

4808 %%% -----
4809 %%% DEFAULT SETS
4810
4811 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4812 \DeclareMicrotypeSetDefault[expansion]{basictext}
4813 \DeclareMicrotypeSetDefault[spacing]{basictext}
4814 \DeclareMicrotypeSetDefault[kerning]{alltext}
4815 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
4816

```

## 15.2 Font variants and aliases

```

4817 %%% -----
4818 %%% FONT VARIANTS AND ALIASES

```

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals):

```

4819
4820 \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.

```

4821
4822 \MT@if@false
4823 \ifx\UnicodeEncodingName\undefined\else
4824   \MT@if@fstreq{\encodingdefault}{\UnicodeEncodingName}\MT@if@true\relax
4825 \fi
4826 \ifMT@fontspec\MT@if@true\fi
4827 \ifMT@if@
4828 \DeclareMicrotypeAlias{lmr}{Latin Modern Roman}
4829   \else
4830 \DeclareMicrotypeAlias{lmr}{cmr}           % lmodern
4831 \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.

```

4832 \DeclareMicrotypeAlias{lmsy}{cmsy}
4833 \DeclareMicrotypeAlias{lmm}{cmm}
4834 \DeclareMicrotypeAlias{aer}{cmr}           % ae
4835 \DeclareMicrotypeAlias{zer}{cmr}           % zefonts
4836 \DeclareMicrotypeAlias{cmor}{cmr}         % eco
4837 \DeclareMicrotypeAlias{hfor}{cmr}         % hfoldsty

```

The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively, also the T<sub>E</sub>X Gyre fonts Pagella and Termes (formerly: qfonts).

```

4838 \DeclareMicrotypeAlias{pxr} {ppl}          % pxfonts
4839 \DeclareMicrotypeAlias{qpl} {ppl}          % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

The 'FPL Neu' fonts, a 're-implementation' of Palatino.
4840 \DeclareMicrotypeAlias{fp9x}{pplx}        % FPL Neu
4841 \DeclareMicrotypeAlias{fp9j}{pplj}        % "

The newpx package, a replacement for pxfonts.
4842 \DeclareMicrotypeAlias{zpllf}{ppl}        % newpxtext
4843 \DeclareMicrotypeAlias{zplosf}{ppl}       % "
4844 \DeclareMicrotypeAlias{zpltlf}{ppl}       % "
4845 \DeclareMicrotypeAlias{zpltosf}{ppl}      % "
4846 \DeclareMicrotypeAlias{txr} {ptm}         % txfonts

The newtx package, a replacement for txfonts.
4847 \DeclareMicrotypeAlias{ntxlf}{ptm}        % newtxtext
4848 \DeclareMicrotypeAlias{ntxosf}{ptm}       % "
4849 \DeclareMicrotypeAlias{ntxtlf}{ptm}       % "
4850 \DeclareMicrotypeAlias{ntxtosf}{ptm}      % "

The tempora package.
4851 \DeclareMicrotypeAlias{Tempora-TLF}{ptm}  % tempora
4852 \DeclareMicrotypeAlias{Tempora-TOf}{ptm} % "
4853 \DeclareMicrotypeAlias{qtm} {ptm}        % TeX Gyre Termes (formerly: qfonts/QuasiTimes)

The OpenType versions:
4854 \DeclareMicrotypeAlias{TeX Gyre Pagella}{Palatino Linotype}
4855 \DeclareMicrotypeAlias{Palatino LT Std} {Palatino Linotype}
4856 \DeclareMicrotypeAlias{Palatino}        {Palatino Linotype}
4857 \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.
4858 \DeclareMicrotypeAlias{zeur}{eur}        % Euler VM
4859 \DeclareMicrotypeAlias{zeus}{eus}        % "

MicroPress's Charter version (chmath).
4860 \DeclareMicrotypeAlias{chr} {bch}        % CH Math

The XCharter package extends the Charter fonts.
4861 \DeclareMicrotypeAlias{XCharter-TLF} {bch} % XCharter
4862 \DeclareMicrotypeAlias{XCharter-TOf}{bch} % "

The mathdesign package provides math fonts matching Bitstream Charter and URW
Garamond.
4863 \DeclareMicrotypeAlias{mdbch}{bch}       % mathdesign/Charter
4864 \DeclareMicrotypeAlias{mdugm}{ugm}       % mathdesign/URW Garamond

The garamondx package, an extension of URW Garamond, providing small caps and
oldstyle figures.
4865 \DeclareMicrotypeAlias{zgmX}{ugm}        % garamondx
4866 \DeclareMicrotypeAlias{zgmj}{ugm}       % "
4867 \DeclareMicrotypeAlias{zgmI}{ugm}       % "
4868 \DeclareMicrotypeAlias{zgmq}{ugm}       % "

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the config-
uration.
4869 \DeclareMicrotypeAlias{ulg} {blg}        % URW LetterGothic -> Bitstream LetterGothic12Pitch

Euro symbol fonts, to save some files.
4870 \DeclareMicrotypeAlias{zpeus} {zpeu}     % Adobe Euro sans -> serif
4871 \DeclareMicrotypeAlias{eurosans}{zpeu}   % Adobe Euro sans -> serif
4872 \DeclareMicrotypeAlias{euroitcs}{euroitc}% ITC Euro sans -> serif
4873

```

### 15.3 Interaction with babel

Contexts that are to be set when switching to a language.

```

4874 %%% -----
4875 %%% INTERACTION WITH THE `babel' PACKAGE
4876
4877 \DeclareMicrotypeBabelHook
4878   {english,UKenglish,british,USenglish,american}
4879   {kerning=, spacing=nonfrench}
4880
4881 \DeclareMicrotypeBabelHook
4882   {french,français,acadian,canadien}
4883   {kerning=french, spacing=}
4884
4885 \DeclareMicrotypeBabelHook
4886   {turkish}
4887   {kerning=turkish, spacing=}
4888

```

### 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  $\text{Lua}_{\LaTeX}$ , 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.

```

4889 </m-t>
4890 <*m-t|zpeu|mys>
4891 %%% -----
4892 %%% CHARACTER INHERITANCE
4893

```

```
4894 </m-t|zpeu|mvs>
4895 <*m-t>
```

### 15.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 (‘fi’ ligature), 013 (‘fi’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```
4896 \DeclareCharacterInheritance
4897   { encoding = OT1 }
4898   { f = {011}, % ff
4899     i = {\i},
4900     j = {\j},
4901     O = {\O},
4902     o = {\o}
4903   }
4904
```

### 15.5.2 T1

Candidates here: 028 (‘fi’), 029 (‘fi’), 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.

```
4905 \DeclareCharacterInheritance
4906   { encoding = T1 }
4907   { A = {\^A,\'A,\^A,\-A,\"A,\r A,\k A,\u A},
4908     a = {\^a,\'a,\^a,\-a,\"a,\r a,\k a,\u a},
4909     C = {\'C,\c C,\v C},
4910     c = {\'c,\c c,\v c},
4911     D = {\v D,\DH},
4912     d = {\v d,\dj},
4913     E = {\^E,\'E,\^E,\^E,\"E,\k E,\v E},
4914     e = {\^e,\'e,\^e,\^e,\"e,\k e,\v e},
4915     f = {027}, % ff
4916     G = {\u G},
4917     g = {\u g},
4918     I = {\^I,\'I,\^I,\^I,\"I,\.I},
4919     i = {\^i,\'i,\^i,\^i,\"i,\i},
4920     j = {\j},
4921     L = {\L,\'L,\v L},
4922     l = {\l,\'l,\v l},
4923     N = {\'N,\-N,\v N},
4924     n = {\'n,\-n,\v n},
4925     O = {\O,\^O,\'O,\^O,\-O,\"O,\H O},
4926     o = {\o,\^o,\'o,\^o,\-o,\"o,\H o},
4927     R = {\'R,\v R},
4928     r = {\'r,\v r},
4929     S = {\'S,\c S,\v S,\SS},
4930     s = {\'s,\c s,\v s},
4931     T = {\c T,\v T},
4932     t = {\c t,\v t},
4933     U = {\^U,\'U,\^U,\^U,\"U,\H U,\r U},
4934     u = {\^u,\'u,\^u,\^u,\"u,\H u,\r u},
4935     Y = {\'Y,\"Y},
4936     y = {\'y,\"y},
4937     Z = {\'Z,\.Z,\v Z},
4938     z = {\'z,\.z,\v z}
```

The ‘soft hyphen’ often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```
4939 % - = {127},
4940 }
4941
```

**15.5.3 LY1**

More characters: 008 ('fl'), 012 ('fi'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

4942 \DeclareCharacterInheritance
4943   { encoding = LY1 }
4944   { A = {\^A,\'A,\^A,\-A,\"A,\r A},
4945     a = {\^a,\'a,\^a,\-a,\"a,\r a},
4946     C = {\c C},
4947     c = {\c c},
4948     D = {\DH},
4949     E = {\^E,\'E,\^E,\"E},
4950     e = {\^e,\'e,\^e,\"e},
4951     f = {011}, % ff
4952     I = {\^I,\'I,\^I,\"I},
4953     i = {\^i,\'i,\^i,\"i,\i},
4954     L = {\L},
4955     l = {\l},
4956     N = {\-N},
4957     n = {\-n},
4958     O = {\^O,\'O,\^O,\-O,\"O,\O},
4959     o = {\^o,\'o,\^o,\-o,\"o,\o},
4960     S = {\v S},
4961     s = {\v s},
4962     U = {\^U,\'U,\^U,\"U},
4963     u = {\^u,\'u,\^u,\"u},
4964     Y = {\'Y,\"Y},
4965     y = {\'y,\"y},
4966     Z = {\v Z},
4967     z = {\v z}
4968   }
4969
```

**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.

```

4970 \DeclareCharacterInheritance
4971   { encoding = OT4 }
4972   { A = {\k A},
4973     a = {\k a},
4974     C = {\'C},
4975     c = {\'c},
4976     E = {\k E},
4977     e = {\k e},
4978     f = {011}, % ff
4979     i = {\i},
4980     j = {\j},
4981     L = {\L},
4982     l = {\l},
4983     N = {\'N},
4984     n = {\'n},
4985     O = {\O,\"O},
4986     o = {\o,\"o},
4987     S = {\'S},
4988     s = {\'s},
4989     Z = {\'Z,\"Z},
4990     z = {\'z,\"z}
4991   }
4992
```

**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.

```

4993 \DeclareCharacterInheritance
4994 { encoding = QX }
4995 { A = {\^A,\'A,\^A,\-A,\"A,\k A,\AA},
4996   a = {\`a,\'a,\^a,\-a,\"a,\k a,\aa},
4997   C = {\'C,\c C},
4998   c = {\'c,\c c},
4999   D = {\DH},
5000   E = {\^E,\'E,\^E,\"E,\k E},
5001   e = {\`e,\'e,\^e,\"e,\k e},
5002   f = {011}, % ff
5003   I = {\^I,\'I,\^I,\"I,\k I},
5004   i = {\`i,\'i,\^i,\"i,\k i,\i},
5005   j = {\j},
5006   L = {\L},
5007   l = {\l},
5008   N = {\'N,\-N},
5009   n = {\'n,\-n},
5010   O = {\0,\^0,\'0,\^0,\-0,\"0},
5011   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.

```

5012   S = {\'S,\c S,\textcommabelow S,\v S},
5013   s = {\'s,\c s,\textcommabelow s,\v s},
5014   T = {\c T,\textcommabelow T},
5015   t = {\c t,\textcommabelow t},
5016   U = {\^U,\'U,\^U,\"U,\k U},
5017   u = {\`u,\'u,\^u,\"u,\k u},
5018   Y = {\'Y,\"Y},
5019   y = {\'y,\"y},
5020   Z = {\'Z,\.Z,\v Z},
5021   z = {\'z,\.z,\v z},
5022   . = \textellipsis
5023 }
5024

```

**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.

```

5025 \DeclareCharacterInheritance
5026 { encoding = T5 }
5027 { A = {\^A,\'A,\-A,\h A,\d A,\^A,\u A,
5028   \^Acircumflex,\'Acircumflex,\-Acircumflex,\hAcircumflex,\dAcircumflex,
5029   \^Abreve,\'Abreve,\-Abreve,\hAbreve,\dAbreve},
5030   a = {\`a,\'a,\-a,\h a,\d a,\^a,\u a,
5031   \^acircumflex,\'acircumflex,\-acircumflex,\hacircumflex,\dacircumflex,
5032   \^abreve,\'abreve,\-abreve,\h\abreve,\d\abreve},
5033   D = {\DJ},
5034   d = {\dj},
5035   E = {\^E,\'E,\-E,\h E,\d E,\^E,
5036   \^Ecircumflex,\'Ecircumflex,\-Ecircumflex,\hEcircumflex,\dEcircumflex},
5037   e = {\`e,\'e,\-e,\h e,\d e,\^e,
5038   \^ecircumflex,\'ecircumflex,\-ecircumflex,\hecircumflex,\decircumflex},

```

<sup>16</sup> Contributed by *Maciej Eder*.

<sup>17</sup> Cf. <http://tug.org/pipermail/tex-live/2008-August/017204.html>

```

5039 I = {\^I,\'I,\-I,\h I,\d I},
5040 i = {\^i,\'i,\-i,\h i,\d i,\i},
5041 O = {\^O,\'O,\-O,\h O,\d O,\^O,\horn O,
5042   \^Ocircumflex,\'Ocircumflex,\-Ocircumflex,\hOcircumflex,\dOcircumflex,
5043   \^Ohorn,\'Ohorn,\-Ohorn,\hOhorn,\dOhorn},
5044 o = {\^o,\'o,\-o,\h o,\d o,\^o,\horn o,
5045   \^ocircumflex,\'ocircumflex,\-ocircumflex,\hocircumflex,\docircumflex,
5046   \^ohorn,\'ohorn,\-ohorn,\hohorn,\dohorn},
5047 U = {\^U,\'U,\-U,\h U,\d U,\horn U,
5048   \^Uhorn,\'Uhorn,\-Uhorn,\hUhorn,\dUhorn},
5049 u = {\^u,\'u,\-u,\h u,\d u,\horn u,
5050   \^uhorn,\'uhorn,\-uhorn,\huhorn,\duhorn},
5051 Y = {\^Y,\'Y,\-Y,\h Y,\d Y},
5052 y = {\^y,\'y,\-y,\h y,\d y}
5053 }
5054

```

### 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.

```

5055 \DeclareCharacterInheritance
5056 { encoding = {EU1,EU2,TU} }
5057 { A = {\^A,\'A,\^A,\-A,\"A,\r A,\k A,\u A},
5058   a = {\^a,\'a,\^a,\-a,\"a,\r a,\k a,\u a},
5059   C = {\'C,\c C,\v C},
5060   c = {\'c,\c c,\v c},
5061   D = {\v D,\DH},
5062   d = {\v d,\dj},
5063   E = {\^E,\'E,\^E,\"E,\k E,\v E},
5064   e = {\^e,\'e,\^e,\"e,\k e,\v e},
5065 %   f = {/f_f}, % sometimes /f_f, sometimes /ff
5066   G = {\u G},
5067   g = {\u g},
5068   I = {\^I,\'I,\^I,\"I,\.I},
5069   i = {\^i,\'i,\^i,\"i,\i},
5070 %   j = {\j},
5071   L = {\L,\'L,\v L},
5072   l = {\l,\'l,\v l},
5073   N = {\'N,\-N,\v N},
5074   n = {\'n,\-n,\v n},
5075   O = {\^O,\'O,\^O,\-O,\"O,\H O},
5076   o = {\^o,\'o,\^o,\-o,\"o,\H o},
5077   R = {\'R,\v R},
5078   r = {\'r,\v r},
5079   S = {\'S,\c S,\v S}, % \SS
5080   s = {\'s,\c s,\v s},
5081   T = {\c T,\v T},
5082   t = {\c t,\v t},
5083   U = {\^U,\'U,\^U,\"U,\H U,\r U},
5084   u = {\^u,\'u,\^u,\"u,\H u,\r u},
5085   Y = {\'Y,\"Y},
5086   y = {\'y,\"y},
5087   Z = {\'Z,\.Z,\v Z},
5088   z = {\'z,\.z,\v z}
5089 }
5090
5091 </m-t>

```

### 15.5.8 Euro symbols

Make Euro symbols settings simpler.

```
5092 < *zpeu >
5093 \DeclareCharacterInheritance
5094   { encoding = U,
5095     family   = {zpeu,zpeus,eurosans} }
5096   { E = 128 }
5097
5098 < /zpeu >
5099 < *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.

```
5100 \DeclareCharacterInheritance
5101   { encoding = {OT1,U},
5102     family   = mvs }
5103   { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
5104
5105 < /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.

```
5106 < *m-t >
5107 %%% -----
5108 %%% TRACKING/LETTERSPPACING
5109
5110 \SetTracking
5111 [ name          = default,
5112   no ligatures = {f} ]
5113 { encoding      = {OT1,T1,T2A,LY1,OT4,QX,EU2,TU} }
5114 { }
5115
```

### 15.7 Font expansion

These are Hàn Thê Thành's original expansion settings. They are used for all fonts (until somebody shows mercy and creates font-specific settings).

```
5116 %%% -----
5117 %%% EXPANSION
5118
5119 \SetExpansion
5120 [ name          = default      ]
5121 { encoding      = {OT1,OT4,QX,T1,LY1} }
5122 {
5123   A = 500,      a = 700,
5124   \AE = 500,   \ae = 700,
5125   B = 700,      b = 700,
5126   C = 700,      c = 700,
5127   D = 500,      d = 700,
5128   E = 700,      e = 700,
5129   F = 700,
5130   G = 500,      g = 700,
5131   H = 700,      h = 700,
5132   K = 700,      k = 700,
5133   M = 700,      m = 700,
5134   N = 700,      n = 700,
5135   O = 500,      o = 700,
```

```

5136  \OE = 500,  \oe = 700,
5137  P = 700,    p = 700,
5138  Q = 500,    q = 700,
5139  R = 700,
5140  S = 700,    s = 700,
5141  U = 700,    u = 700,
5142  W = 700,    w = 700,
5143  Z = 700,    z = 700,
5144  2 = 700,
5145  3 = 700,
5146  6 = 700,
5147  8 = 700,
5148  9 = 700
5149  }
5150

```

### Settings for Cyrillic T2A encoding.<sup>18</sup>

```

5151 \SetExpansion
5152 [ name = T2A ]
5153 { encoding = T2A }
5154 {
5155  A = 500,    a = 700,
5156  B = 700,    b = 700,
5157  C = 700,    c = 700,
5158  D = 500,    d = 700,
5159  E = 700,    e = 700,
5160  F = 700,
5161  G = 500,    g = 700,
5162  H = 700,    h = 700,
5163  K = 700,    k = 700,
5164  M = 700,    m = 700,
5165  N = 700,    n = 700,
5166  O = 500,    o = 700,
5167  P = 700,    p = 700,
5168  Q = 500,    q = 700,
5169  R = 700,
5170  S = 700,    s = 700,
5171  U = 700,    u = 700,
5172  W = 700,    w = 700,
5173  Z = 700,    z = 700,
5174  2 = 700,
5175  3 = 700,
5176  6 = 700,
5177  8 = 700,
5178  9 = 700,
5179  \CYRA = 500,  \cyra = 700,
5180  \CYRB = 700,  \cyrb = 700,
5181  \CYRV = 700,  \cyrv = 700,
5182  \CYRG = 700,  \cyrg = 700,
5183  \CYRD = 700,  \cyrd = 700,
5184  \CYRE = 700,  \cyre = 700,
5185  \CYRZH = 700, \cyrzh = 700,
5186  \CYRZ = 700,  \cyrz = 700,
5187  \CYRI = 700,  \cyri = 700,
5188  \CYRISHRT = 700, \cyrishrt = 700,
5189  \CYRK = 700,  \cyrk = 700,
5190  \CYRL = 700,  \cyrl = 700,
5191  \CYRM = 700,  \cyrm = 700,
5192  \CYRN = 700,  \cyrn = 700,
5193  \CYRO = 500,  \cyro = 700,
5194  \CYRP = 700,  \cyrp = 700,
5195  \CYRR = 700,  \cyrr = 700,
5196  \CYRS = 700,  \cyrs = 700,
5197  \CYRT = 700,  \cyrt = 700,

```

---

18 Contributed by *Karl Karlsson*.

```

5198 \CYRU = 700, \cyru = 700,
5199 \CYRF = 700, \cyrf = 700,
5200 \CYRH = 700, \cyrh = 700,
5201 \CYRC = 700, \cyrc = 700,
5202 \CYRCH = 700, \cyrch = 700,
5203 \CYRSH = 700, \cyrsh = 700,
5204 \CYRSHCH = 700, \cyrshch = 700,
5205 \CYRHRDSN = 700, \cyrhrdsn = 700,
5206 \CYRERY = 700, \cyrery = 700,
5207 \CYRSFTSN = 700, \cyrsoftsn = 700,
5208 \CYREREV = 700, \cyrerev = 700,
5209 \CYRYU = 700, \cyryu = 700,
5210 \CYRYA = 700, \cyrya = 700
5211 }
5212

```

T5 encoding does not contain \AE, \ae, \OE and \oe.

```

5213 \SetExpansion
5214 [ name = T5 ]
5215 { encoding = T5 }
5216 {
5217 A = 500, a = 700,
5218 B = 700, b = 700,
5219 C = 700, c = 700,
5220 D = 500, d = 700,
5221 E = 700, e = 700,
5222 F = 700,
5223 G = 500, g = 700,
5224 H = 700, h = 700,
5225 K = 700, k = 700,
5226 M = 700, m = 700,
5227 N = 700, n = 700,
5228 O = 500, o = 700,
5229 P = 700, p = 700,
5230 Q = 500, q = 700,
5231 R = 700,
5232 S = 700, s = 700,
5233 U = 700, u = 700,
5234 W = 700, w = 700,
5235 Z = 700, z = 700,
5236 2 = 700,
5237 3 = 700,
5238 6 = 700,
5239 8 = 700,
5240 9 = 700
5241 }
5242
5243 </m-t>

```

## 15.8 Character protrusion

```

5244 %%% -----
5245 %%% PROTRUSION
5246

```

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.

```

5247 <*cfg-t>
5248 \SetProtrusion
5249 <m-t> [ name = default ]

```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```
5250 <bch> [ name = bch-default ]
```

- Bitstream Letter Gothic (blg)

```
5251 <blg> [ name = blg-default ]
```

- Computer Modern Roman (cmr)

```
5252 <cmr> [ name = cmr-default ]
```

- Adobe Garamond (pad, padx, padj)

```
5253 <pad> [ name = pad-default ]
```

- Minion<sup>19</sup> (pmnx, pmnj)

```
5254 <pmn> [ name = pmnj-default ]
```

- Palatino (ppl, pplx, pplj)

```
5255 <ppl> [ name = ppl-default ]
```

- Times (ptm, ptmx, ptmj)

```
5256 <ptm> [ name = ptm-default ]
```

- URW Garamond (ugm)

---

19 Contributed by *Harald Harders* and *Karl Karlsson*.

```

5257 <ugm> [ name = ugm-default ]
5258 <m-t|cmr|pmn> { }
5259 <bch|blg|pad|ugm> { encoding = OT1,
5260 <ppl|ptm> { encoding = {OT1,OT4},
5261 <bch> family = bch }
5262 <blg> family = blg }
5263 <pad> family = {pad,padx,padj} }
5264 <ppl> family = {ppl,pplx,pplj} }
5265 <ptm> family = {ptm,ptmx,ptmj} }
5266 <ugm> family = ugm }
5267 {
5268 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> A = {50,50},
5269 <ugm> A = {50,100},
5270 <pad|ptm> \AE = {50, },
5271 <ugm> \AE = {150,50},
5272 <ugm> B = { ,50},
5273 <bch|pad|pmn|ugm> C = {50, },
5274 <bch|pad|pmn> D = { ,50},
5275 <ugm> D = { ,70},
5276 <ugm> E = { ,50},
5277 <m-t|bch|cmr|pad|pmn|ptm> F = { ,50},
5278 <ugm> F = { ,70},
5279 <bch|pad|pmn> G = {50, },
5280 <ugm> G = {50,50},
5281 <blg> I = {150,150},
5282 <m-t|cmr|pad|pmn|ppl|ptm|ugm> J = {50, },
5283 <bch|blg> J = {100, },
5284 <!blg> K = { ,50},
5285 <blg> K = {50, },
5286 <m-t|bch|cmr|pad|pmn|ppl> L = { ,50},
5287 <blg> L = { ,150},
5288 <ptm> L = { ,80},
5289 <ugm> L = { ,120},
5290 <bch|pad|pmn|ugm> O = {50,50},
5291 <pad> \OE = {50, },
5292 <ugm> \OE = {50,50},
5293 <blg> P = { ,100},
5294 <ugm> P = { ,50},
5295 <bch|pad|pmn> Q = {50,70},
5296 <ugm> Q = {50,50},
5297 <bch> R = { ,50},
5298 <ugm> R = { ,70},
5299 <m-t|bch|cmr|pad|pmn|ppl|ptm> T = {50,50},
5300 <blg> T = {100,100},
5301 <ugm> T = {70,70},
5302 <m-t|bch|cmr|pad|pmn|ppl|ptm> V = {50,50},
5303 <blg|ugm> V = {70,70},
5304 <m-t|bch|cmr|pad|pmn|ppl|ptm> W = {50,50},
5305 <ugm> W = {70,70},
5306 <m-t|bch|cmr|pad|pmn|ppl|ptm> X = {50,50},
5307 <ugm> X = {50,70},
5308 <m-t|bch|cmr|pad|pmn|ppl> Y = {50,50},
5309 <blg|ptm|ugm> Y = {80,80},
5310 <ugm> Z = {50,50},
5311 <blg> f = {150,100},
5312 <blg> i = {150,150},
5313 <blg> j = {100,100},
5314 <m-t|bch|cmr|pad|pmn|ppl|ptm> k = { ,50},
5315 <ugm> k = { ,70},
5316 <blg> l = {150,150},
5317 <pmn> l = { , -50},
5318 <pad|ppl> p = {50,50},
5319 <ugm> p = { ,50},
5320 <pad|ppl> q = {50, },
5321 <!blg> r = { ,50},

```

```

5322 <blg>      r = {100, 80},
5323 <cmr|pad|pmn>    t = { ,70},
5324 <bch>      t = { ,50},
5325 <blg>      t = {150, 80},
5326 <ugm>      t = { ,100},
5327 <m-t|bch|cmr|pad|pmn|ppl|ptm>    v = {50,50},
5328 <blg>      v = {100,100},
5329 <ugm>      v = {50,70},
5330 <m-t|bch|cmr|pad|pmn|ppl|ptm>    w = {50,50},
5331 <ugm>      w = {50,70},
5332 <!blg>     x = {50,50},
5333 <blg>      x = {100,100},
5334 <m-t|bch|pad|pmn>    y = { ,50},
5335 <blg>      y = { 50,100},
5336 <cmr|ppl|ptm>    y = {50,70},
5337 <ugm>      y = { ,70},

5338 <cmr>      0 = { ,50},
5339 <m-t>      1 = {50,50},
5340 <bch|blg|pad|ptm|ugm>    1 = {150,150},
5341 <cmr>      1 = {100,200},
5342 <pmn>      1 = { ,50},
5343 <ppl>      1 = {100,100},
5344 <bch|cmr|pad|ugm>    2 = {50,50},
5345 <blg>      2 = { ,100},
5346 <bch|pmn>    3 = {50, },
5347 <cmr|pad|ugm>    3 = {50,50},
5348 <blg>      3 = {100, },
5349 <m-t|pad>    4 = {50,50},
5350 <bch>      4 = {100,50},
5351 <blg>      4 = {100, },
5352 <cmr|ugm>    4 = {70,70},
5353 <pmn>      4 = {50, },
5354 <ptm>      4 = {70, },
5355 <cmr>      5 = { ,50},
5356 <pad>      5 = {50,50},
5357 <bch>      6 = {50, },
5358 <cmr>      6 = { ,50},
5359 <pad>      6 = {50,50},
5360 <m-t>      7 = {50,50},
5361 <bch|pad|pmn|ugm>    7 = {50,80},
5362 <blg>      7 = {100,100},
5363 <cmr|ptm>    7 = {50,100},
5364 <ppl>      7 = { ,50},
5365 <cmr>      8 = { ,50},
5366 <bch|pad>    9 = {50,50},
5367 <cmr>      9 = { ,50},
5368 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    . = { ,700},
5369 <bch>      . = { ,600},
5370 <blg>      . = {400,500},
5371 <!blg>     {,}= { ,500},
5372 <blg>     {,}= {300,400},
5373 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    : = { ,500},
5374 <bch>      : = { ,400},
5375 <blg>      : = {300,400},
5376 <m-t|bch|pad|pmn|ptm>    ; = { ,300},
5377 <blg>      ; = {200,300},
5378 <cmr|ppl>    ; = { ,500},
5379 <ugm>      ; = { ,400},
5380 <!blg>     ! = { ,100},
5381 <blg>      ! = {200,200},
5382 <m-t|pad|pmn|ptm>    ? = { ,100},
5383 <bch|cmr|ppl|ugm>    ? = { ,200},
5384 <blg>      ? = {150,150},
5385 <pmn>      " = {300,300},
5386 <m-t|bch|cmr|pad|pmn|ppl>    @ = {50,50},

```

```

5387 <ptm> @ = {100,100},
5388 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> ~ = {200,250},
5389 <ugm> ~ = {300,350},
5390 <pad|ppl|ptm> & = {50,100},
5391 <ugm> & = { ,100},
5392 <m-t|cmr|pad|pmn> \% = {50,50},
5393 <bch> \% = { ,50},
5394 <ppl|ptm> \% = {100,100},
5395 <ugm> \% = {50,100},
5396 <blg> \# = {100,100},
5397 <m-t|ppl|ptm|ugm> * = {200,200},
5398 <bch|pmn> * = {200,300},
5399 <blg> * = {150,200},
5400 <cmr|pad> * = {300,300},
5401 <m-t|cmr|ppl|ptm> + = {250,250},
5402 <bch> + = {150,250},
5403 <pad> + = {300,300},
5404 <blg|pmn> + = {150,200},
5405 <ugm> + = {250,300},
5406 <blg|ugm> {=} = {200,200},
5407 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
5408 <bch|ugm> ( = {200, }, ) = { ,200},
5409 <cmr|blg> ( = {300, }, ) = { ,300},
5410 <ppl> ( = {100, }, ) = { ,300},
5411 <bch|pmn> [ = {100, }, ] = { ,100},
5412 <blg> [ = {300,100}, ] = { ,300},

5413 <m-t|pad|pmn|ptm> / = {100,200},
5414 <bch> / = { ,200},
5415 <blg> / = {300,300},
5416 <cmr|ppl> / = {200,300},
5417 <ugm> / = {100,300},
5418 <m-t|ptm> - = {500,500},
5419 <bch|cmr|ppl> - = {400,500},
5420 <blg> - = {300,400},
5421 <pad> - = {300,500},
5422 <pmn> - = {200,400},
5423 <ugm> - = {500,600},
5424 <blg> < = {200,100}, > = {100,200},
5425 <blg> _ = {150,250},
5426 <blg> | = {250,250},
5427 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
5428 <bch> \textendash = {200,300}, \textendash = {150,250},
5429 <cmr> \textendash = {400,300}, \textendash = {300,200},
5430 <pad|ppl|ptm> \textendash = {300,300}, \textendash = {200,200},
5431 <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).

```

5432 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
5433 <blg> \textquoteleft = {400,600}, \textquoteright = {400,600},
5434 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
5435 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
5436 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
5437 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
5438 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300}
5439 <blg> \textquotedblright = {300,400}
5440 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600}
5441 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400}
5442 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400}
5443 }
5444

```

Greek uppercase letters are in OT1 encoding only.

```

5445 <*m-t|cmr|pmn>

```

```

5446 \SetProtrusion
5447 <m-t> [ name = OT1-default,
5448 <cmr> [ name = cmr-OT1,
5449 <pmn> [ name = pmnj-OT1,
5450 <m-t> load = default ]
5451 <cmr> load = cmr-default ]
5452 <pmn> load = pmnj-default ]
5453 <m-t> { encoding = OT1 }
5454 <cmr> { encoding = {OT1,OT4},
5455 <pmn> { encoding = OT1,
5456 <cmr> family = cmr }
5457 <pmn> family = pmnj }
5458 {
5459 <m-t|cmr> \AE = {50, },
5460 <pmn> \OE = {50, }
5461 <*cmr>
5462 "00 = { ,150}, % \Gamma
5463 "01 = {100,100}, % \Delta
5464 "02 = { 50, 50}, % \Theta
5465 "03 = {100,100}, % \Lambda
5466 "06 = { 50, 50}, % \Sigma
5467 "07 = {100,100}, % \Upsilon
5468 "08 = { 50, 50}, % \Phi
5469 "09 = { 50, 50} % \Psi

```

Remaining slots can be found in the source file.

```

5470 </cmr>
5471 }
5472
5473 </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).

```

5474 \SetProtrusion
5475 <m-t> [ name = T1-default,
5476 <bch> [ name = bch-T1,
5477 <blg> [ name = blg-T1,
5478 <cmr> [ name = cmr-T1,
5479 <pad> [ name = pad-T1,
5480 <pmn> [ name = pmnj-T1,
5481 <ppl> [ name = ppl-T1,
5482 <ptm> [ name = ptm-T1,
5483 <ugm> [ name = ugm-T1,
5484 <m-t> load = default ]
5485 <bch> load = bch-default ]
5486 <blg> load = blg-default ]
5487 <cmr> load = cmr-default ]
5488 <pad> load = pad-default ]
5489 <pmn> load = pmnj-default ]
5490 <ppl> load = ppl-default ]
5491 <ptm> load = ptm-default ]
5492 <ugm> load = ugm-default ]
5493 <m-t> { encoding = {T1,LY1,EU1,EU2,TU} }
5494 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
5495 <blg|ptm|ugm> { encoding = {T1},
5496 <bch> family = bch }
5497 <blg> family = blg }
5498 <cmr> family = cmr }
5499 <pad> family = {pad,padx,padj} }
5500 <pmn> family = pmnj }
5501 <ppl> family = {ppl,pplx,pplj} }
5502 <ptm> family = {ptm,ptmx,ptmj} }
5503 <ugm> family = ugm }
5504 {

```

```

5505 <m-t|cmr> \AE = {50, },
5506 <bch|pmn> \OE = {50, },
5507 <pmn> \TH = { ,50},
5508 <blg> \v L = { ,250},
5509 <blg> \v d = { ,250},
5510 <blg> \v l = { ,250},
5511 <blg> \v t = { ,250},
5512 <blg> 127 = {300,400},
5513 <blg> 156 = {100, }, % IJ
5514 <blg> 188 = { 80, 80}, % ij
5515 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
5516 <cmr> _ = {200,200},
5517 <ugm> _ = {100,200},
5518 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
5519 <bch> \textbackslash = {150,200},
5520 <blg> \textbackslash = {250,300},
5521 <cmr|ppl> \textbackslash = {200,300},
5522 <ugm> \textbackslash = {100,300},
5523 <ugm> \textbar = {200,200},
5524 <blg> \textendash = {300,300}, \textemdash = {150,150},
5525 <blg> \textquotedbl = {300,400}, \textquotedblleft = {300,400},
5526 <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.

```

5527 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
5528 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
5529 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
5530 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsingright = {300,400},
5531 <blg> \guilsinglleft = {300,500}, \guilsingright = {300,500},
5532 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsingright = {300,500},
5533 <ugm> \guilsinglleft = {400,400}, \guilsingright = {300,600},
5534 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5535 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5536 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
5537 <blg|pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
5538 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
5539 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
5540 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
5541 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
5542 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
5543 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
5544 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
5545 <pmn> \textless = {100, }, \textgreater = { ,100},
5546 <pmn> \textvisiblespace = {100,100} % not in LY1

5547 }
5548

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```

5549 <*cmr>
5550 \SetProtrusion
5551 [ name = lmr-T1,
5552 load = cmr-T1 ]
5553 { encoding = {T1,LY1},
5554 family = lmr }
5555 {
5556 \textquotedblleft = {300,400}, \textquotedblright = {300,400}
5557 }
5558
5559 </cmr>

```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).<sup>20</sup>

```

5560 <*m-t|cmr|pmn)
5561 \SetProtrusion
5562 <m-t) [ name = T2A-default,
5563 <cmr) [ name = cmr-T2A,
5564 <pmn) [ name = pmnj-T2A,
5565 <m-t) load = default ]
5566 <cmr) load = cmr-default ]
5567 <pmn) load = pmnj-default ]
5568 { encoding = T2A,
5569 <m-t) }
5570 <cmr) family = cmr }
5571 <pmn) family = pmnj }
5572 {
5573 \CYRA = {50,50},
5574 \CYRG = { ,50},
5575 \CYRK = { ,50},
5576 \CYRT = {50,50},
5577 \CYRH = {50,50},
5578 \CYRU = {50,50},
5579 <pmn) \CYRS = {50, },
5580 <pmn) \CYRO = {50,50},
5581 \cyrk = { ,50},
5582 \cyrg = { ,50},
5583 \cyrh = {50,50},
5584 <m-t|pmn) \cyru = {50,50},
5585 <cmr) \cyru = {50,70},
5586 <m-t) - = {100,100},
5587 <cmr) - = {200,200},
5588 <m-t) \textbackslash = {100,200}, \quotedblbase = {400,400},
5589 <cmr) \textbackslash = {200,300}, \quotedblbase = {400,400},
5590 <pmn) \textbackslash = {100,200}, \quotedblbase = {300,300},
5591 <cmr) \textquotedbl = {300,300}, \textquotedblleft = {200,600},
5592 <m-t) \guillemotleft = {200,200}, \guillemotright = {200,200},
5593 <cmr) \guillemotleft = {300,200}, \guillemotright = {100,400},
5594 <pmn) \guillemotleft = {200,200}, \guillemotright = {150,300},
5595 <m-t|cmr) \textbraceleft = {400,200}, \textbraceright = {200,400},
5596 <pmn) \textbraceleft = {200, }, \textbraceright = { ,300},
5597 <m-t|cmr) \textless = {200,100}, \textgreater = {100,200}
5598 <pmn) \textless = {100, }, \textgreater = { ,100}
5599 }
5600
5601 </m-t|cmr|pmn)

```

Settings for the QX encoding (generic and Times).<sup>21</sup> It also includes some glyphs otherwise in TS1.

```

5602 <*m-t|ptm)
5603 \SetProtrusion
5604 <m-t) [ name = QX-default,
5605 <ptm) [ name = ptm-QX,
5606 <m-t) load = default ]
5607 <ptm) load = ptm-default ]
5608 <m-t) { encoding = QX }
5609 <ptm) { encoding = QX,
5610 <ptm) family = {ptm,ptmx,ptmj} }
5611 {
5612 \AE = {50, },
5613 <ptm) * = {200,200},
5614 {=} = {100,100},
5615 \textunderscore = {100,100},
5616 \textbackslash = {100,200},
5617 <ptm) \quotedblbase = {400,400},

```

---

20 Contributed by Karl Karlsson.

21 Contributed by Maciej Eder.

```

5618 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5619 <ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
5620 \textexclamdown = {100, }, \textquestiondown = {100, },
5621 <m-t> \textbraceleft = {400,200}, \textbraceright = {200,400},
5622 <ptm> \textbraceleft = {200,200}, \textbraceright = {200,300},
5623 \textless = {200,100}, \textgreater = {100,200},
5624 \textminus = {200,200}, \textdegree = {300,300},
5625 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5626 <ptm> \copyright = {100,150}, \textregistered = {100,150},
5627 <ptm> \textxgeq = { ,100}, \textxleq = {100, },
5628 <ptm> \textalpha = { , 50}, \textDelta = { 70, 70},
5629 <ptm> \textpi = { 50, 80}, \textSigma = { , 70},
5630 <ptm> \textmu = { , 80}, \texteuro = { 50, 50},
5631 <ptm> \textellipsis = {150,200}, \textasciitilde = { 80, 80},
5632 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
5633 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5634 <ptm> \textdiv = { 50,150}, \textsection = { 80, 80},
5635 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
5636 <ptm> \textbullet = {150,150}, \textperiodcentered = {300,300},
5637 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5638 <ptm> \textperthousand = { ,50}
5639 }
5640
5641 </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.

```

5642 <*cmr|bch>
5643 \SetProtrusion
5644 <cmr> [ name = cmr-T5,
5645 <cmr> load = cmr-default ]
5646 <bch> [ name = bch-T5,
5647 <bch> load = bch-default ]
5648 { encoding = T5,
5649 <cmr> family = cmr }
5650 <bch> family = bch }
5651 {
5652 <bch> _ = {100,100},
5653 <bch> \textbackslash = {150,200},
5654 <cmr> \textbackslash = {200,300},
5655 <cmr> \textquotedblleft = {200,600},
5656 <cmr> \textquotedbl = {300,300},
5657 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
5658 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
5659 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
5660 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5661 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},
5662 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5663 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
5664 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
5665 \textless = {200,100}, \textgreater = {100,200}
5666 }
5667
5668 </cmr|bch>

```

Minion with lining numbers.

```

5669 <*pmn>
5670 \SetProtrusion
5671 [ name = pmnx-OT1,
5672 load = pmnj-default ]
5673 { encoding = OT1,
5674 family = pmnx }
5675 {
5676 1 = {230,180}
5677 }

```

```

5678
5679 \SetProtrusion
5680 [ name = pmnx-T1,
5681   load = pmnj-T1 ]
5682 { encoding = {T1,LY1},
5683   family = pmnx }
5684 {
5685   1 = {230,180}
5686 }
5687
5688 \SetProtrusion
5689 [ name = pmnx-T2A,
5690   load = pmnj-T2A ]
5691 { encoding = {T2A},
5692   family = pmnx }
5693 {
5694   1 = {230,180}
5695 }
5696
5697 </pmn>

```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```

5698 < *ptm >
5699 \SetProtrusion
5700 [ name = ptm-LY1,
5701   load = ptm-T1 ]
5702 { encoding = LY1,
5703   family = {ptm,ptmx,ptmj} }
5704 {
5705   - = {100,100},
5706   \texttrademark = {100,100},
5707   \textregistered = {100,100},
5708   \textcopyright = {100,100},
5709   \textdegree = {300,300},
5710   \textminus = {200,200},
5711   \textellipsis = {150,200},
5712 % \texteuro = { , }, % ?
5713   \textcent = {100,100},
5714   \textquotesingle = {500,500},
5715   \textflorin = { 50, 70},
5716   \textdagger = {150,150},
5717   \textdaggerdbl = {100,100},
5718   \textperthousand = { , 50},
5719   \textbullet = {150,150},
5720   \textonesuperior = {100,100},
5721   \texttwosuperior = { 50, 50},
5722   \textthreesuperior = { 50, 50},
5723   \textperiodcentered = {300,300},
5724   \textplusminus = { 50, 80},
5725   \textmultiply = {100,100},
5726   \textdivide = { 50,150}

```

Remaining slots in the source file.

```

5727   }
5728 }
5729 </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>

```

5730 \SetProtrusion
5731 <m-t> [ name = OT1-it ]
5732 <bch> [ name = bch-it ]
5733 <blg> [ name = blg-it,
5734 <blg> load = blg-default ]
5735 <cmr> [ name = cmr-it ]
5736 <pad> [ name = pad-it ]
5737 <pmn> [ name = pmnj-it ]
5738 <ppl> [ name = ppl-it ]
5739 <ptm> [ name = ptm-it ]
5740 <ugm> [ name = ugm-it ]
5741 <m-t|bch|blg|pad|ugm> { encoding = OT1,
5742 <ppl|ptm> { encoding = {OT1,OT4},
5743 <bch> family = bch,
5744 <blg> family = blg,
5745 <pad> family = {pad,padx,padj},
5746 <ppl> family = {ppl,pplx,pplj},
5747 <ptm> family = {ptm,ptmx,ptmj},
5748 <ugm> family = ugm,
5749 <m-t|bch|pad|ppl|ptm> shape = {it,sl} }
5750 <blg|ugm> shape = it }
5751 <cmr|pmn> { }
5752 {
5753 <cmr> A = {100,100},
5754 <ptm> A = {100,50},
5755 <pad|pmn> A = {50, },
5756 <ugm> A = { ,150},
5757 <ppl> A = {50,50},
5758 <ptm> \AE = {100, },
5759 <pad|ppl> \AE = {50, },
5760 <cmr> B = {83,-40},
5761 <pad|ppl|ptm> B = {50, },
5762 <pmn> B = {20,-50},
5763 <bch|ppl|ptm|ugm> C = {50, },
5764 <cmr> C = {165,-75},
5765 <pad> C = {100, },
5766 <pmn> C = {50,-50},
5767 <cmr> D = {75, -28},
5768 <pad|ppl|ptm> D = {50,50},
5769 <pmn> D = {20, },
5770 <cmr> E = {80,-55},
5771 <pad|ppl|ptm> E = {50, },
5772 <pmn> E = {20,-50},
5773 <cmr> F = {85,-80},
5774 <pad|ptm> F = {100, },
5775 <pmn> F = {10, },
5776 <ppl> F = {50, },
5777 <bch|ppl|ptm|ugm> G = {50, },
5778 <cmr> G = {153,-15},
5779 <pad> G = {100, },
5780 <pmn> G = {50,-50},
5781 <cmr> H = {73,-60},
5782 <pad|ppl|ptm> H = {50, },
5783 <cmr> I = {140,-120},
5784 <pad|ptm> I = {50, },
5785 <pmn> I = {20,-50},
5786 <cmr> J = {135,-80},
5787 <pad> J = {50, },
5788 <pmn> J = {20, },

```

22 Settings contributed by Hendrik Vogt.

5789 <ptm> J = {100, },  
5790 <cmr> K = {70,-30},  
5791 <pad|ppl|ptm> K = {50, },  
5792 <pmn> K = {20, },  
5793 <cmr> L = {87, 40},  
5794 <pad|ppl|ptm> L = {50, },  
5795 <pmn> L = {20,50},  
5796 <ugm> L = { ,100},  
5797 <cmr> M = {67,-45},  
5798 <pmn> M = { , -30},  
5799 <ptm> M = {50, },  
5800 <cmr> N = {75,-55},  
5801 <pmn> N = { , -30},  
5802 <ptm> N = {50, },  
5803 <bch|pmn|ppl|ptm> O = {50, },  
5804 <cmr> O = {150,-30},  
5805 <pad> O = {100, },  
5806 <ugm> O = {70,50},  
5807 <ppl|ptm> \OE = {50, },  
5808 <pad> \OE = {100, },  
5809 <cmr> P = {82,-50},  
5810 <pad|ppl|ptm> P = {50, },  
5811 <pmn> P = {20,-50},  
5812 <bch|pmn|ppl|ptm> Q = {50, },  
5813 <cmr> Q = {150,-30},  
5814 <pad> Q = {100, },  
5815 <ugm> Q = {70,50},  
5816 <cmr> R = {75, 15},  
5817 <pad|ppl|ptm> R = {50, },  
5818 <pmn> R = {20, },  
5819 <bch|pad|ppl|ptm> S = {50, },  
5820 <cmr> S = {90,-65},  
5821 <pmn> S = {20,-30},  
5822 <bch|pad|ppl|ptm> \$ = {50, },  
5823 <cmr> \$ = {100,-20},  
5824 <pmn> \$ = {20,-30},  
5825 <bch|pmn|ugm> T = {70, },  
5826 <cmr> T = {220,-85},  
5827 <pad|ppl|ptm> T = {100, },  
5828 <cmr> U = {230,-55},  
5829 <pad|ppl|ptm> U = {50, },  
5830 <pmn> U = {50,-50},  
5831 <cmr> V = {260,-60},  
5832 <pad|pmn|ugm> V = {100, },  
5833 <ppl|ptm> V = {100,50},  
5834 <cmr> W = {185,-55},  
5835 <pad|pmn|ugm> W = {100, },  
5836 <ppl> W = {50, },  
5837 <ptm> W = {100,50},  
5838 <cmr> X = {70,-30},  
5839 <ppl|ptm> X = {50, },  
5840 <cmr> Y = {250,-60},  
5841 <pmn> Y = {50, },  
5842 <ppl> Y = {100,50},  
5843 <ptm> Y = {100, },  
5844 <cmr> Z = {90,-60},  
5845 <pmn> Z = { , -50},  
5846 <cmr> a = {150,-10},  
5847 <cmr> b = {170, },  
5848 <cmr> c = {173,-10},  
5849 <cmr> d = {150,-55},  
5850 <pmn> d = { , -50},  
5851 <cmr> e = {180, },  
5852 <cmr> f = { , -250},  
5853 <pad|pmn> f = { , -100},

```

5854 <cmr> g = {150,-10},
5855 <cmr> h = {100, },
5856 <cmr> i = {210, },
5857 <pmn> i = { , -30},
5858 <cmr> j = { , -40},
5859 <pmn> j = { , -30},
5860 <cmr> k = {110,-50},
5861 <cmr> l = {240,-110},
5862 <pmn> l = { , -100},
5863 <cmr> m = {80, },
5864 <cmr> n = {115, },
5865 <bch> o = {50,50},
5866 <cmr> o = {155, },
5867 <bch> p = { , 50},
5868 <pmn> p = {-50, },
5869 <bch> q = {50, },
5870 <cmr> q = {170,-40},
5871 <cmr> r = {155,-40},
5872 <pmn> r = { , 50},
5873 <cmr> s = {130, },
5874 <bch> t = { , 50},
5875 <cmr> t = {230,-10},
5876 <cmr> u = {120, },
5877 <cmr> v = {140,-25},
5878 <pmn|ugm> v = {50, },
5879 <bch> w = { , 50},
5880 <cmr> w = {98,-20},
5881 <pmn|ugm> w = {50, },
5882 <cmr> x = {65,-40},
5883 <bch> y = { , 50},
5884 <cmr> y = {130,-20},
5885 <cmr> z = {110,-80},
5886 <cmr> 0 = {170,-85},
5887 <bch|ptm> 1 = {150,100},
5888 <cmr> 1 = {230,110},
5889 <pad> 1 = {150, },
5890 <pmn> 1 = {50, },
5891 <ppl> 1 = {100, },
5892 <ugm> 1 = {150,150},
5893 <cmr> 2 = {130,-70},
5894 <pad|ppl|ptm> 2 = {50, },
5895 <pmn> 2 = {-50, },
5896 <bch> 3 = {50, },
5897 <cmr> 3 = {140,-70},
5898 <pmn> 3 = {-100, },
5899 <ptm> 3 = {100,50},
5900 <bch> 4 = {100, },
5901 <cmr> 4 = {130,80},
5902 <pad> 4 = {150, },
5903 <ppl|ptm> 4 = {50, },
5904 <cmr> 5 = {160, },
5905 <ptm> 5 = {50, },
5906 <bch> 6 = {50, },
5907 <cmr> 6 = {175,-30},
5908 <bch|pad|ptm> 7 = {100, },
5909 <cmr> 7 = {250,-150},
5910 <pmn> 7 = {20, },
5911 <ppl> 7 = {50, },
5912 <cmr> 8 = {130,-40},
5913 <cmr> 9 = {155,-80},
5914 <m-t|cmr|pad|pmn|ppl> . = { , 500},
5915 <blg> . = {400,600},
5916 <bch|ptm|ugm> . = { , 700},
5917 <blg> {,} = {300,500},
5918 <m-t|pad|pmn|ppl> {,} = { , 500},

```

```

5919 <cmr> {,}= { ,450},
5920 <bch|ugm> {,}= { ,600},
5921 <ptm> {,}= { ,700},
5922 <m-t|cmr|pad|ppl> := { ,300},
5923 <bch|ugm> := { ,400},
5924 <pmn> := { ,200},
5925 <ptm> := { ,500},
5926 <m-t|cmr|pad|ppl> ; = { ,300},
5927 <bch|ugm> ; = { ,400},
5928 <pmn> ; = { ,200},
5929 <ptm> ; = { ,500},
5930 <ptm> != { ,100},
5931 <bch> ? = { ,200},
5932 <ptm> ? = { ,100},
5933 <ppl> ? = { ,300},
5934 <pmn> " = {400,200},
5935 <m-t|pad|pmn|ppl|ptm> & = {50,50},
5936 <bch> & = { ,80},
5937 <cmr> & = {130,30},
5938 <ugm> & = {50,100},
5939 <m-t|pad|pmn> \% = {100, },
5940 <cmr> \% = {180,50},
5941 <bch> \% = {50,50},
5942 <ppl|ptm> \% = {100,100},
5943 <ugm> \% = {100,50},
5944 <m-t|pmn|ppl> * = {200,200},
5945 <bch> * = {300,200},
5946 <cmr> * = {380,20},
5947 <pad> * = {500,100},
5948 <ptm|ugm> * = {400,200},
5949 <m-t|pmn|ppl> + = {150,200},
5950 <cmr> + = {180,200},
5951 <bch|ugm> + = {250,250},
5952 <pad|ptm> + = {250,200},
5953 <m-t|pad|pmn|ppl> @ = {50,50},
5954 <bch> @ = {80,50},
5955 <cmr> @ = {180,10},
5956 <ptm> @ = {150,150},
5957 <m-t|bch|ugm> ~ = {150,150},
5958 <cmr|pad|pmn|ppl|ptm> ~ = {200,150},
5959 <ugm> {=} = {200,200},
5960 <m-t|bch|pad|pmn|ppl|ptm|ugm> ( = {200, }, ) = { ,200},
5961 <cmr> ( = {300, }, ) = { ,70},
5962 <m-t|pad|ppl|ptm|ugm> / = {100,200},
5963 <cmr> / = {100,100},
5964 <bch> / = { ,150},
5965 <pmn> / = {100,150},
5966 <m-t> - = {300,300},
5967 <bch|pad> - = {300,400},
5968 <pmn> - = {200,300},
5969 <cmr> - = {500,300},
5970 <ppl> - = {300,500},
5971 <ptm> - = {500,500},
5972 <ugm> - = {400,700},
5973 <blg> - = {0,300},
5974 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
5975 <bch> \textendash = {200,300}, \textendash = {150,200},
5976 <cmr> \textendash = {500,300}, \textendash = {400,170},
5977 <pad|ppl|ptm|ugm> \textendash = {300,300}, \textendash = {200,200},
5978 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
5979 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
5980 <cmr> \textquoteleft = {800,200}, \textquoteright = {800,-20},
5981 <pad> \textquoteleft = {800,200}, \textquoteright = {800,200},
5982 <ppl> \textquoteleft = {700,400}, \textquoteright = {700,400},
5983 <ptm> \textquoteleft = {800,500}, \textquoteright = {800,500},

```

```

5984 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
5985 <blg> \textquotedblright = {300,300}
5986 <cmr> \textquotedblleft = {540,100}, \textquotedblright = {500,100}
5987 <pad> \textquotedblleft = {700,200}, \textquotedblright = {700,200}
5988 <ppl> \textquotedblleft = {500,300}, \textquotedblright = {500,300}
5989 <ptm> \textquotedblleft = {700,400}, \textquotedblright = {700,400}
5990 <ugm> \textquotedblleft = {600,200}, \textquotedblright = {600,200}
5991 }
5992
5993 <*cmr|pmn>
5994 \SetProtrusion
5995 <cmr> [ name = cmr-it-OT1,
5996 <pmn> [ name = pmnj-it-OT1,
5997 <cmr> [ load = cmr-it ]
5998 <pmn> [ load = pmnj-it ]
5999 <cmr> { encoding = {OT1,OT4},
6000 <pmn> { encoding = OT1,
6001 <cmr> family = cmr,
6002 <pmn> family = pmnj,
6003 <cmr> shape = it }
6004 <pmn> shape = {it,s1} }
6005 {
6006 <cmr> \AE = {100, },
6007 <pmn> \AE = { , -50},
6008 <cmr> \OE = {100, },
6009 <pmn> \OE = {50, }
6010 <*cmr>
6011 "00 = {200,150}, % \Gamma
6012 "01 = {150,100}, % \Delta
6013 "02 = {150, 50}, % \Theta
6014 "03 = {150, 50}, % \Lambda
6015 "04 = {100,100}, % \Xi
6016 "05 = {100,100}, % \Pi
6017 "06 = {100, 50}, % \Sigma
6018 "07 = {200,150}, % \Upsilon
6019 "08 = {150, 50}, % \Phi
6020 "09 = {150,100}, % \Psi
6021 "0A = { 50, 50} % \Omega
6022 </cmr>
6023 }
6024
6025 </cmr|pmn>
6026 \SetProtrusion
6027 <m-t> [ name = T1-it-default,
6028 <bch> [ name = bch-it-T1,
6029 <blg> [ name = blg-it-T1,
6030 <cmr> [ name = cmr-it-T1,
6031 <pad> [ name = pad-it-T1,
6032 <pmn> [ name = pmnj-it-T1,
6033 <ppl> [ name = ppl-it-T1,
6034 <ptm> [ name = ptm-it-T1,
6035 <ugm> [ name = ugm-it-T1,
6036 <m-t> [ load = OT1-it ]
6037 <bch> [ load = bch-it ]
6038 <blg> [ load = blg-T1 ]
6039 <cmr> [ load = cmr-it ]
6040 <pmn> [ load = pmnj-it ]
6041 <pad> [ load = pad-it ]
6042 <ppl> [ load = ppl-it ]
6043 <ptm> [ load = ptm-it ]
6044 <ugm> [ load = ugm-it ]
6045 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
6046 <blg|ptm|ugm> { encoding = T1,
6047 <bch> family = bch,
6048 <blg> family = blg,

```

```

6049 <cmr>    family = cmr,
6050 <pmn>    family = pmnj,
6051 <pad>    family = {pad,padx,padj},
6052 <ppl>    family = {ppl,pplx,pplj},
6053 <ptm>    family = {ptm,ptmx,ptmj},
6054 <ugm>    family = ugm,
6055 <m-t|bch|pad|pmn|ppl|ptm> shape = {it,sl} }
6056 <blg|cmr|ugm> shape = it }
6057 {
6058 <m-t|bch|pmn>    _ = { ,100},
6059 <blg>           _ = {0,300},
6060 <cmr|ugm>       _ = {100,200},
6061 <pad|ppl|ptm>  _ = {100,100},
6062 <blg>           . = {400,600},
6063 <blg>           {,}= {300,500},
6064 <cmr>           \AE = {100, } ,
6065 <pmn>           \AE = { , -50},
6066 <bch|pmn>       \OE = { 50, },
6067 <cmr>           \OE = {100, } ,
6068 <pmn>           O31 = { , -100}, % ffl
6069 <cmr|ptm>       156 = {100, }, % IJ
6070 <pad>           156 = {50, }, % IJ
6071 <pmn>           156 = {20, }, % IJ
6072 <pmn>           188 = { , -30}, % ij
6073 <pmn>           \v t = { ,100},
6074 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
6075 <cmr|ugm>       \textbackslash = {300,300},
6076 <bch>           \textbackslash = {150,150},
6077 <pmn>           \textbackslash = {100,150},
6078 <ugm>           \textbar = {200,200},
6079 <cmr>           \textquotedblleft = {500,300},
6080 <blg>           \textquotedleft = {400,400}, \textquoteright = {400,400},
6081 <blg>           \textquotedbl = {300,300}, \textquotedblleft = {300,300},
6082 <blg>           \textquotedblright = {300,300}, \quotedblbase = {200,600},
6083 <m-t|ptm>       \quotesinglbase = {300,700}, \quotedblbase = {400,500},
6084 <cmr>           \quotesinglbase = {300,700}, \quotedblbase = {200,600},
6085 <bch|pmn>       \quotesinglbase = {200,500}, \quotedblbase = {150,500},
6086 <pad|ppl>       \quotesinglbase = {500,500}, \quotedblbase = {400,400},
6087 <ugm>           \quotesinglbase = {300,700}, \quotedblbase = {300,500},
6088 <m-t|ppl|ptm>  \guilsinglleft = {400,400}, \guilsingright = {300,500},
6089 <bch|pmn>       \guilsinglleft = {300,400}, \guilsingright = {200,500},
6090 <cmr>           \guilsinglleft = {500,300}, \guilsingright = {400,400},
6091 <pad>           \guilsinglleft = {500,400}, \guilsingright = {300,500},
6092 <ugm>           \guilsinglleft = {400,400}, \guilsingright = {300,600},
6093 <m-t|ppl>       \guillemotleft = {300,300}, \guillemotright = {300,300},
6094 <bch|pmn>       \guillemotleft = {200,300}, \guillemotright = {150,400},
6095 <cmr>           \guillemotleft = {400,100}, \guillemotright = {200,300},
6096 <pad>           \guillemotleft = {300,300}, \guillemotright = {200,400},
6097 <ptm>           \guillemotleft = {300,400}, \guillemotright = {200,400},
6098 <ugm>           \guillemotleft = {300,400}, \guillemotright = {300,400},
6099 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
6100 <cmr|ptm>       \textexclamdown = {200, }, \textquestiondown = {200, },
6101 <pmn>           \textexclamdown = {-50, }, \textquestiondown = {-50, },
6102 <m-t|ppl|ugm>  \textbraceleft = {200,100}, \textbraceright = {200,200},
6103 <bch|pmn>       \textbraceleft = {200, }, \textbraceright = { ,200},
6104 <cmr|pad|ptm>  \textbraceleft = {400,100}, \textbraceright = {200,200},
6105 <bch|pmn>       \textless = {100, }, \textgreater = { ,100},
6106 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
6107 <pmn>           \textvisiblespace = {100,100}
6108 }
6109
6110 <*m-t|cmr|pmn>
6111 \SetProtrusion
6112 <m-t> [ name = T2A-it-default,
6113 <cmr> [ name = cmr-it-T2A,

```

```

6114 <pmn> [ name      = pmnj-it-T2A,
6115 <m-t>   load      = OT1-it  ]
6116 <cmr>   load      = cmr-it  ]
6117 <pmn>   load      = pmnj-it  ]
6118   { encoding = T2A,
6119 <cmr>   family   = cmr,
6120 <pmn>   family   = pmnj,
6121 <m-t|pmn> shape   = {it,s1} }
6122 <cmr>   shape   = it      }
6123   {
6124 <cmr>   \CYRA = {100,50},
6125 <pmn>   \CYRA = {50, },
6126 <cmr>   \CYRB = {50, },
6127 <cmr>   \CYRV = {50, },
6128 <pmn>   \CYRV = {20,-50},
6129 <cmr>   \CYRG = {100, },
6130 <pmn>   \CYRG = {10, },
6131 <cmr>   \CYRD = {50, },
6132 <cmr>   \CYRE = {50, },
6133 <pmn>   \CYRE = {20,-50},
6134 <cmr>   \CYRZH = {50, },
6135 <cmr>   \CYRZ = {50, },
6136 <pmn>   \CYRZ = {20,-50},
6137 <cmr>   \CYRI = {50, },
6138 <pmn>   \CYRI = { , -30},
6139 <cmr>   \CYRISHRT = {50, },
6140 <cmr>   \CYRK = {50, },
6141 <pmn>   \CYRK = {20, },
6142 <cmr>   \CYRL = {50, },
6143 <cmr>   \CYRM = {50, },
6144 <pmn>   \CYRM = { , -30},
6145 <cmr>   \CYRN = {50, },
6146 <cmr>   \CYRO = {100, },
6147 <pmn>   \CYRO = {50, },
6148 <cmr>   \CYRP = {50, },
6149 <cmr>   \CYRR = {50, },
6150 <pmn>   \CYRR = {20,-50},
6151 <cmr>   \CYRS = {100, },
6152 <pmn>   \CYRS = {50, },
6153 <cmr>   \CYRT = {100, },
6154 <pmn>   \CYRT = {70, },
6155 <cmr>   \CYRU = {100, },
6156 <pmn>   \CYRU = {50, },
6157 <cmr>   \CYRF = {100, },
6158 <cmr>   \CYRH = {50, },
6159 <cmr>   \CYRC = {50, },
6160 <cmr>   \CYRCH = {100, },
6161 <cmr>   \CYRSH = {50, },
6162 <cmr>   \CYRSHCH = {50, },
6163 <cmr>   \CYRHRDSN = {100, },
6164 <cmr>   \CYRERY = {50, },
6165 <cmr>   \CYRSFTSN = {50, },
6166 <cmr>   \CYREREV = {50, },
6167 <cmr>   \CYRYU = {50, },
6168 <cmr>   \CYRYA = {50, },
6169 <pmn>   \CYRYA = { , 20},
6170 <pmn>   \cyrr = {-50, },
6171 <m-t|pmn>   _ = { , 100},
6172 <cmr>   _ = {100,200},
6173 <pmn>   031 = { , -100}, % ff1
6174 <pmn>   \v t = { , 100},
6175 <m-t>   \textbackslash = {100,200}, \quotedblbase = {400,500},
6176 <cmr>   \textbackslash = {300,300}, \quotedblbase = {200,600},
6177 <pmn>   \textbackslash = {100,150}, \quotedblbase = {150,500},
6178 <m-t>   \guillemotleft = {300,300}, \guillemotright = {300,300},

```

```

6179 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
6180 <pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
6181 <m-t> \textbraceleft = {200,100}, \textbraceright = {200,200},
6182 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
6183 <pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
6184 <cmr> \textquotedblleft = {500,300},
6185 <cmr> \textless = {300,100}, \textgreater = {200,100}
6186 <pmn> \textless = {100, }, \textgreater = { ,100}
6187 }
6188
6189 </m-t|cmr|pmn>
6190 < *m-t|ptm>
6191 \SetProtrusion
6192 <m-t> [ name = QX-it-default,
6193 <ptm> [ name = ptm-it-QX,
6194 <m-t> load = OT1-it ]
6195 <ptm> load = ptm-it ]
6196 { encoding = {QX},
6197 <ptm> family = {ptm,ptmx,ptmj},
6198 shape = {it,sI} }
6199 {
6200 <ptm> 009 = { , 50}, % fk
6201 {=} = {100,100},
6202 <m-t> \textunderscore = {100,100},
6203 <ptm> \textunderscore = {100,150},
6204 \textbackslash = {100,200},
6205 \quotedblbase = {300,400},
6206 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
6207 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
6208 \textexclamdown = {200, }, \textquestiondown = {200, },
6209 \textbraceleft = {200,100}, \textbraceright = {200,200},
6210 \textless = {100,100}, \textgreater = {100,100},
6211 \textminus = {200,200}, \textdegree = {300,150},
6212 <m-t> \copyright = {100,100}, \textregistered = {100,100}
6213 <ptm> \textregistered = {100,150}, \copyright = {100,150},
6214 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
6215 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
6216 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
6217 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
6218 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
6219 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
6220 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
6221 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
6222 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
6223 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
6224 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
6225 <ptm> \textperthousand = { ,50}
6226 }
6227
6228 </m-t|ptm>
6229 < *cmr|bch>
6230 \SetProtrusion
6231 <cmr> [ name = cmr-it-T5,
6232 <cmr> load = cmr-it ]
6233 <bch> [ name = bch-it-T5,
6234 <bch> load = bch-it ]
6235 { encoding = T5,
6236 <bch> family = bch,
6237 <cmr> family = cmr,
6238 shape = it }
6239 {
6240 <bch> _ = { ,100},
6241 <cmr> _ = {100,200},
6242 <bch> \textbackslash = {150,150},
6243 <cmr> \textbackslash = {300,300},

```

```

6244 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
6245 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
6246 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
6247 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
6248 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
6249 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
6250 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
6251 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
6252 <bch> \textless = {100, }, \textgreater = { ,100},
6253 <cmr> \textless = {300,100}, \textgreater = {200,100}
6254 }
6255
6256 </cmr|bch>

```

Slanted is very similar to italic.

```

6257 <*cmr>
6258 \SetProtrusion
6259 [ name = cmr-sl,
6260 load = cmr-it-OT1 ]
6261 { encoding = {OT1,OT4},
6262 family = cmr,
6263 shape = sl }
6264 {
6265 L = { ,50},
6266 f = { ,-50},
6267 - = {300, },
6268 \textendash = {400, }, \textemdash = {300, }
6269 }
6270
6271 \SetProtrusion
6272 [ name = cmr-sl-T1,
6273 load = cmr-it-T1 ]
6274 { encoding = {T1,LY1},
6275 family = cmr,
6276 shape = sl }
6277 {
6278 L = { ,50},
6279 f = { ,-50},
6280 - = {300, },
6281 \textendash = {400, }, \textemdash = {300, }
6282 }
6283
6284 \SetProtrusion
6285 [ name = cmr-sl-T2A,
6286 load = cmr-it-T2A ]
6287 { encoding = T2A,
6288 family = cmr,
6289 shape = sl }
6290 {
6291 L = { ,50},
6292 f = { ,-50},
6293 - = {300, },
6294 \textendash = {400, }, \textemdash = {300, }
6295 }
6296
6297 \SetProtrusion
6298 [ name = cmr-sl-T5,
6299 load = cmr-it-T5 ]
6300 { encoding = T5,
6301 family = cmr,
6302 shape = sl }
6303 {
6304 L = { ,50},
6305 f = { ,-50},
6306 - = {300, },

```

```

6307     \textendash = {400, }, \textemdash = {300, }
6308   }
6309
6310 \SetProtrusion
6311 [ name = lmr-it-T1,
6312   load = cmr-it-T1 ]
6313 { encoding = {T1,LY1},
6314   family = lmr,
6315   shape = {it,s1} }
6316 {
6317   \textquotedblleft = { ,200}, \textquotedblright = { ,200},
6318   \quotesinglbase = { ,400}, \quotedblbase = { ,500}
6319 }
6320

```

Oldstyle numerals are slightly different.

```

6321 \SetProtrusion
6322 [ name = cmr(oldstyle)-it,
6323   load = cmr-it-T1 ]
6324 { encoding = T1,
6325   family = {hfor,cmor},
6326   shape = {it,s1} }
6327 {
6328   1 = {250, 50},
6329   2 = {150,-100},
6330   3 = {100,-50},
6331   4 = {150,150},
6332   6 = {200, },
6333   7 = {200, 50},
6334   8 = {150,-50},
6335   9 = {100, 50}
6336 }
6337
6338 </cmr>
6339 < *pmn >
6340 \SetProtrusion
6341 [ name = pmnx-it,
6342   load = pmnj-it ]
6343 { encoding = OT1,
6344   family = pmnx,
6345   shape = {it,s1} }
6346 {
6347   1 = {100,150}
6348 }
6349
6350 \SetProtrusion
6351 [ name = pmnx-it-T1,
6352   load = pmnj-it-T1 ]
6353 { encoding = {T1,LY1},
6354   family = pmnx,
6355   shape = {it,s1} }
6356 {
6357   1 = {100,150}
6358 }
6359
6360 \SetProtrusion
6361 [ name = pmnx-it-T2A,
6362   load = pmnj-it-T2A ]
6363 { encoding = {T2A},
6364   family = pmnx,
6365   shape = {it,s1} }
6366 {
6367   1 = {100,150}
6368 }
6369

```

```

6370 </pmn>
6371 <*ptm>
6372 \SetProtrusion
6373 [ name = ptm-it-LY1,
6374   load = ptm-it-T1 ]
6375 { encoding = {LY1},
6376   family = {ptm,ptmx,ptmj},
6377   shape = {it,s1} }
6378 {
6379   - = {100,100},
6380   \texttrademark = {100,100},
6381   \textregistered = {100,100},
6382   \textcopyright = {100,100},
6383   \textdegree = {300,100},
6384   \textminus = {200,200},
6385   \textellipsis = {100,200},
6386   % \texteuro = { , }, % ?
6387   \textcent = {100,100},
6388   \textquotesingle = {500, },
6389   \textflorin = {100, 70},
6390   \textdagger = {150,150},
6391   \textdaggerdbl = {100,100},
6392   \textbullet = {150,150},
6393   \textonesuperior = {150,100},
6394   \texttwosuperior = {150, 50},
6395   \textthreesuperior = {150, 50},
6396   \textparagraph = {100, },
6397   \textperiodcentered = {500,300},
6398   \textonequarter = { 50, },
6399   \textonehalf = { 50, },
6400   \textplusminus = {100,100},
6401   \textmultiply = {150,150},
6402   \textdivide = {150,150}
6403 }
6404
6405 </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).

```

6406 <*(blg|ugm)>
6407 \SetProtrusion
6408 <m-t> [ name = OT1-sc,
6409 <bch> [ name = bch-sc,
6410 <cmr> [ name = cmr-sc-OT1,
6411 <pad> [ name = pad-sc,
6412 <pmn> [ name = pmnj-sc,
6413 <ppl> [ name = ppl-sc,
6414 <ptm> [ name = ptm-sc,
6415 <m-t> load = default ]
6416 <bch> load = bch-default ]
6417 <cmr> load = cmr-OT1 ]
6418 <pad> load = pad-default ]
6419 <pmn> load = pmnj-default ]
6420 <ppl> load = ppl-default ]
6421 <ptm> load = ptm-default ]
6422 <m-t|bch|pad|pmn> { encoding = OT1,
6423 <cmr|ppl|ptm> { encoding = {OT1,OT4},
6424 <bch> family = bch,
6425 <cmr> family = cmr,
6426 <pad> family = {pad,padx,padj},

```

```

6427 <pmn>    family = pmnj,
6428 <ppl>    family = {ppl,pplx,pplj},
6429 <ptm>    family = {ptm,ptmx,ptmj},
6430    shape = sc }
6431  {
6432    a = {50,50},
6433 <cmr|pad|ppl|ptm> \ae = {50, },
6434 <bch|pmn>    c = {50, },
6435 <bch|pad|pmn> d = { ,50},
6436 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
6437 <bch|pad|pmn> g = {50, },
6438 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
6439 <bch>    j = {100, },
6440 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
6441 <ptm>    l = { ,80},
6442 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
6443 <ptm>    013 = { ,80}, % fl
6444 <bch|pad|pmn> o = {50,50},
6445 <pad|pmn> \oe = {50, },
6446 <ppl>    p = { 0, 0},
6447 <bch|pad|pmn> q = {50,70},
6448 <ppl>    q = { 0, },
6449 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
6450    t = {50,50},
6451 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
6452 <ptm>    y = {80,80}
6453  }
6454
6455 \SetProtrusion
6456 <m-t> [ name = T1-sc,
6457 <bch> [ name = bch-sc-T1,
6458 <cmr> [ name = cmr-sc-T1,
6459 <pad> [ name = pad-sc-T1,
6460 <pmn> [ name = pmnj-sc-T1,
6461 <ppl> [ name = ppl-sc-T1,
6462 <ptm> [ name = ptm-sc-T1,
6463 <m-t> load = T1-default ]
6464 <bch> load = bch-T1 ]
6465 <cmr> load = cmr-T1 ]
6466 <pad> load = pad-T1 ]
6467 <pmn> load = pmnj-T1 ]
6468 <ppl> load = ppl-T1 ]
6469 <ptm> load = ptm-T1 ]
6470  { encoding = {T1,LY1},
6471 <bch> family = bch,
6472 <cmr> family = cmr,
6473 <pad> family = {pad,padx,padj},
6474 <pmn> family = pmnj,
6475 <ppl> family = {ppl,pplx,pplj},
6476 <ptm> family = {ptm,ptmx,ptmj},
6477    shape = sc }
6478  {
6479    a = {50,50},
6480 <cmr|pad|ppl|ptm> \ae = {50, },
6481 <bch|pmn>    c = {50, },
6482 <bch|pad|pmn> d = { ,50},
6483 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
6484 <bch|pad|pmn> g = {50, },
6485 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
6486 <bch>    j = {100, },
6487 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
6488 <ptm>    l = { ,80},
6489 <m-t|bch|cmr|pad|pmn|ppl> 029 = { ,50}, % fl
6490 <ptm>    029 = { ,80}, % fl
6491 <bch|pad|pmn> o = {50,50},

```

```

6492 <bch|pad|pmn> \oe = {50, },
6493 <ppl> p = { 0, 0},
6494 <bch|pad|pmn> q = {50,70},
6495 <ppl> q = { 0, },
6496 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
6497 t = {50,50},
6498 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
6499 <ptm> y = {80,80}
6500 }
6501
6502 <!!(blg|ugm)>
6503 <*m-t|cmr>
6504 \SetProtrusion
6505 <m-t> [ name = T2A-sc,
6506 <cmr> [ name = cmr-sc-T2A,
6507 <m-t> load = T2A-default ]
6508 <cmr> load = cmr-T2A ]
6509 { encoding = T2A,
6510 <cmr> family = cmr,
6511 shape = sc }
6512 {
6513 \cyra = {50,50},
6514 \cyrg = { ,50},
6515 \cyrt = {50,50},
6516 \cyry = { ,50}
6517 }
6518
6519 </m-t|cmr>
6520 <*m-t>
6521 \SetProtrusion
6522 [ name = QX-sc,
6523 load = QX-default ]
6524 { encoding = QX,
6525 shape = sc }
6526 {
6527 a = {50,50},
6528 f = { ,50},
6529 j = {50, },
6530 l = { ,50},
6531 O13 = { ,50}, % fl
6532 r = { , 0},
6533 t = {50,50},
6534 y = {50,50}
6535 }
6536
6537 </m-t>
6538 <*cmr|bch>
6539 \SetProtrusion
6540 <bch> [ name = bch-sc-T5,
6541 <bch> load = bch-T5 ]
6542 <cmr> [ name = cmr-sc-T5,
6543 <cmr> load = cmr-T5 ]
6544 { encoding = T5,
6545 <bch> family = bch,
6546 <cmr> family = cmr,
6547 shape = sc }
6548 {
6549 a = {50,50},
6550 <bch> c = {50, },
6551 <bch> d = { ,50},
6552 f = { ,50},
6553 <bch> g = {50, },
6554 <bch> j = {100, },
6555 <cmr> j = {50, },
6556 l = { ,50},

```

```

6557 <bch>    o = {50,50},
6558 <bch>    q = { 0,  },
6559 <cmr>    r = {  , 0},
6560        t = {50,50},
6561        y = {50,50}
6562    }
6563
6564 </cmr|bch>
6565 <*pmn>
6566 \SetProtrusion
6567 [ name    = pmnx-sc,
6568   load    = pmnj-sc ]
6569 { encoding = OT1,
6570   family  = pmnx,
6571   shape   = sc }
6572 {
6573   1 = {230,180}
6574 }
6575
6576 \SetProtrusion
6577 [ name    = pmnx-sc-T1,
6578   load    = pmnj-sc-T1 ]
6579 { encoding = {T1,LY1},
6580   family  = pmnx,
6581   shape   = sc }
6582 {
6583   1 = {230,180}
6584 }
6585

```

#### 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`.

```

6586 \SetProtrusion
6587 [ name    = pmnj-scit,
6588   load    = pmnj-it  ]
6589 { encoding = OT1,
6590   family  = pmnj,
6591   shape   = {scit,si} }
6592 {
6593   a = {50,  },
6594   \ae = {  , -50},
6595   b = {20, -50},
6596   c = {50, -50},
6597   d = {20, 0},
6598   e = {20, -50},
6599   f = {10, 0},
6600   012 = {10, -50}, % fi
6601   013 = {10, -50}, % fl
6602   014 = {10, -50}, % ffi
6603   015 = {10, -50}, % ffl
6604   g = {50, -50},
6605   i = {20, -50},
6606   j = {20, 0},
6607   k = {20,  },
6608   l = {20, 50},
6609   m = {  , -30},
6610   n = {  , -30},
6611   o = {50,  },
6612   \oe = {50, -50},
6613   p = {20, -50},
6614   q = {50,  },
6615   r = {20, 0},

```

```

6616     s = {20,-30},
6617     t = {70, },
6618     u = {50,-50},
6619     v = {100, },
6620     w = {100, },
6621     y = {50, },
6622     z = { , -50}
6623 }
6624
6625 \SetProtrusion
6626 [ name   = pmnj-scit-T1,
6627   load   = pmnj-it-T1 ]
6628 { encoding = {T1,LY1},
6629   family  = pmnj,
6630   shape   = {scit,si} }
6631 {
6632   a = {50, },
6633   \ae = { , -50},
6634   b = {20,-50},
6635   c = {50,-50},
6636   d = {20, 0},
6637   e = {20,-50},
6638   f = {10, 0},
6639   028 = {10,-50}, % fi
6640   029 = {10,-50}, % fl
6641   030 = {10,-50}, % ffi
6642   031 = {10,-50}, % ffl
6643   g = {50,-50},
6644   i = {20,-50},
6645   188 = {20, 0}, % ij
6646   j = {20, 0},
6647   k = {20, },
6648   l = {20,50},
6649   m = { , -30},
6650   n = { , -30},
6651   o = {50, },
6652   \oe = {50,-50},
6653   p = {20,-50},
6654   q = {50, },
6655   r = {20, 0},
6656   s = {20,-30},
6657   t = {70, },
6658   u = {50,-50},
6659   v = {100, },
6660   w = {100, },
6661   y = {50, },
6662   z = { , -50}
6663 }
6664
6665 \SetProtrusion
6666 [ name   = pmnx-scit,
6667   load   = pmnj-scit ]
6668 { encoding = OT1,
6669   family  = pmnx,
6670   shape   = {scit,si} }
6671 {
6672   l = {100,150}
6673 }
6674
6675 \SetProtrusion
6676 [ name   = pmnx-scit-T1,
6677   load   = pmnj-scit-T1 ]
6678 { encoding = {T1,LY1},
6679   family  = pmnx,
6680   shape   = {scit,si} }

```

```

6681 {
6682   1 = {100,150}
6683 }
6684
6685 (/pmn)

```

### 15.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

6686 \SetProtrusion
6687 <m-t> [ name = textcomp ]
6688 <bch> [ name = bch-textcomp ]
6689 <blg> [ name = blg-textcomp ]
6690 <cmr> [ name = cmr-textcomp ]
6691 <pad> [ name = pad-textcomp ]
6692 <pmn> [ name = pmn-textcomp ]
6693 <ppl> [ name = ppl-textcomp ]
6694 <ptm> [ name = ptm-textcomp ]
6695 <ugm> [ name = ugm-textcomp ]
6696 <m-t> { encoding = TS1 }
6697 <!m-t> { encoding = TS1,
6698   <bch> family = bch }
6699   <blg> family = blg }
6700   <cmr> family = cmr }
6701   <pad> family = {pad,padx,padj} }
6702   <pmn> family = {pmnx,pmnj} }
6703   <ppl> family = {ppl,pplx,pplj} }
6704   <ptm> family = {ptm,ptmx,ptmj} }
6705   <ugm> family = ugm }
6706 {
6707   <blg> \textquotestraightbase = {400,500},
6708   <cmr> \textquotestraightbase = {300,300},
6709   <pad|pmn> \textquotestraightbase = {400,400},
6710   <blg> \textquotestraightdblbase = {300,400},
6711   <cmr|pmn> \textquotestraightdblbase = {300,300},
6712   <pad> \textquotestraightdblbase = {400,400},
6713   <bch|cmr|pad|pmn|ugm> \texttwelveudash = {200,200},
6714   <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
6715   <ugm> \textthreequartersemdash = {200,200},
6716   <blg> \textquotesingle = {500,600},
6717   <cmr|pmn> \textquotesingle = {300,400},
6718   <pad> \textquotesingle = {400,500},
6719   <ptm> \textquotesingle = {500,500},
6720   <ugm> \textquotesingle = {300,500},
6721   <bch|cmr|pmn> \textasteriskcentered = {200,300},
6722   <blg> \textasteriskcentered = {150,200},
6723   <pad> \textasteriskcentered = {300,300},
6724   <ugm> \textasteriskcentered = {100,200},
6725   <pmn> \textfrownstyle = {-200,-200},
6726   <cmr> \textoneoldstyle = {100,100},
6727   <pmn> \textoneoldstyle = { , 50},
6728   <cmr> \textthreeoldstyle = { , 50},
6729   <pad|pmn> \textthreeoldstyle = { 50, },
6730   <cmr> \textfouroldstyle = { 50, 50},
6731   <pad|pmn> \textfouroldstyle = { 50, },
6732   <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
6733   <cmr> \textlangle = {400, },
6734   <cmr> \textrightangle = { ,400},
6735   <m-t|bch|pmn|ptm> \textminus = {200,200},
6736   <cmr|pad|ppl> \textminus = {300,300},
6737   <blg|ugm> \textminus = {250,300},
6738   <bch|pad|pmn> \textlbrackdbl = {100, },
6739   <blg> \textlbrackdbl = {200, },

```

```

6740 <bch|pad|pmn> \textrbrackdbl = { ,100},
6741 <blg> \textrbrackdbl = { ,200},
6742 <pmn> \textasciigrave = {200,500},
6743 <bch|blg|cmr|pad|pmn> \texttildebelow = {200,250},
6744 <pmn> \textasciibreve = {300,400},
6745 <pmn> \textasciicaron = {300,400},
6746 <pmn> \textacutedbl = {200,300},
6747 <pmn> \textgravedbl = {150,300},
6748 <bch|pmn|ugm> \textdagger = { 80, 80},
6749 <blg> \textdagger = {200,200},
6750 <cmr|pad> \textdagger = {100,100},
6751 <ptm> \textdagger = {150,150},
6752 <blg> \textdaggerdbl = {150,150},
6753 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
6754 <ptm> \textdaggerdbl = {100,100},
6755 <bch> \textbardbl = {100,100},
6756 <blg|ugm> \textbardbl = {150,150},
6757 <bch> \textbullet = {200,200},
6758 <blg> \textbullet = {400,500},
6759 <cmr|pad|pmn> \textbullet = { ,100},
6760 <ptm> \textbullet = {150,150},
6761 <ugm> \textbullet = { 50,100},
6762 <bch|cmr|pmn> \textcelsius = { 50, },
6763 <pad> \textcelsius = { 80, },
6764 <bch> \textflorin = { 50, 50},
6765 <blg> \textflorin = {100,100},
6766 <pad|ugm> \textflorin = { ,100},
6767 <pmn> \textflorin = { 50,100},
6768 <ptm> \textflorin = { 50, 70},
6769 <cmr> \textcolonmonetary = { , 50},
6770 <pad|pmn> \textcolonmonetary = { 50, },
6771 <pmn> \textinterrobang = { ,100},
6772 <pmn> \textinterrobangdown = {100, },
6773 <m-t|pad|ptm> \texttrademark = {100,100},
6774 <bch> \texttrademark = {150,150},
6775 <blg|cmr|ppl> \texttrademark = {200,200},
6776 <pmn> \texttrademark = { 50, 50},
6777 <ugm> \texttrademark = {100,150},
6778 <bch|ugm> \textcent = { 50, },
6779 <ptm> \textcent = {100,100},
6780 <bch> \textsterling = { 50, },
6781 <ugm> \textsterling = { , 50},
6782 <bch> \textbrokenbar = {200,200},
6783 <blg> \textbrokenbar = {250,250},
6784 <ugm> \textbrokenbar = {200,300},
6785 <pmn> \textasciidieresis = {300,400},
6786 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
6787 <pmn> \textcopyright = {100,150},
6788 <ppl> \textcopyright = {200,200},
6789 <bch|cmr|ugm> \textordfeminine = {100,200},
6790 <pad|pmn> \textordfeminine = {200,200},
6791 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
6792 <blg> \textlnot = {200,100},
6793 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
6794 <pmn> \textregistered = { 50,150},
6795 <ppl> \textregistered = {200,200},
6796 <pmn> \textasciimacron = {150,200},
6797 <m-t|ppl|ptm> \textdegree = {300,300},
6798 <bch> \textdegree = {150,200},
6799 <blg|ugm> \textdegree = {200,200},
6800 <cmr|pad> \textdegree = {400,400},
6801 <pmn> \textdegree = {150,400},
6802 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
6803 <blg> \textpm = {100,100},
6804 <ptm> \textpm = { 50, 80},

```

```

6805 <bch|blg|ugm> \texttwosuperior = {100,200},
6806 <cmr> \texttwosuperior = { 50,100},
6807 <pad|pmn> \texttwosuperior = {200,200},
6808 <ptm> \texttwosuperior = { 50, 50},
6809 <bch|blg|ugm> \textthreesuperior = {100,200},
6810 <cmr> \textthreesuperior = { 50,100},
6811 <pad|pmn> \textthreesuperior = {200,200},
6812 <ptm> \textthreesuperior = { 50, 50},
6813 <pmn> \textasciicute = {300,400},
6814 <bch|ugm> \textmu = { ,100},
6815 <bch|pad|pmn> \textparagraph = { ,100},
6816 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
6817 <blg> \textperiodcentered = {400,500},
6818 <ptm> \textperiodcentered = {300,300},
6819 <ugm> \textperiodcentered = {200,500},
6820 <bch|blg|ugm> \textonesuperior = {200,300},
6821 <cmr|pad|pmn> \textonesuperior = {200,200},
6822 <ptm> \textonesuperior = {100,100},
6823 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
6824 <blg|cmr> \textordmasculine = {100,200},
6825 <bch|cmr|pmn> \texteuro = {100, },
6826 <pad> \texteuro = { 50,100},
6827 <bch> \texttimes = {200,200},
6828 <blg|ptm> \texttimes = {100,100},
6829 <cmr> \texttimes = {150,250},
6830 <pad> \texttimes = {100,150},
6831 <pmn> \texttimes = { 70,100},
6832 <ugm> \texttimes = {200,300},
6833 <bch|pad|pmn> \textdiv = {150,200}
6834 <blg> \textdiv = {100,100}
6835 <cmr> \textdiv = {150,250}
6836 <ptm> \textdiv = { 50,100},
6837 <ugm> \textdiv = {200,300},
6838 <ptm> \textperthousand = { ,50}
6839 <ugm> \textsection = { ,100},
6840 <ugm> \textonehalf = { 50,100},
6841 <ugm> \textonequarter = { 50,100},
6842 <ugm> \textthreequarters = { 50,100},
6843 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

6844 }
6845
6846 <*cmr|pad|pmn|ugm>
6847 \SetProtrusion
6848 <cmr> [ name = cmr-textcomp-it ]
6849 <pad> [ name = pad-textcomp-it ]
6850 <pmn> [ name = pmn-textcomp-it ]
6851 <ugm> [ name = ugm-textcomp-it ]
6852 { encoding = TS1,
6853 <cmr> family = cmr,
6854 <pad> family = {pad,padx,padj},
6855 <pmn> family = {pmnx,pmnj},
6856 <ugm> family = ugm,
6857 <!ugm> shape = {it,sl} }
6858 <ugm> shape = it }
6859 {
6860 <cmr> \textquotestraightbase = {300,600},
6861 <pad|pmn> \textquotestraightbase = {400,400},
6862 <cmr> \textquotestraightdblbase = {300,600},
6863 <pad> \textquotestraightdblbase = {300,400},
6864 <pmn> \textquotestraightdblbase = {300,300},
6865 \texttwelveudash = {200,200},
6866 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
6867 <ugm> \textthreequartersemdash = {200,200},

```

```

6868 <cmr> \textquotesingle = {600,300},
6869 <pad> \textquotesingle = {800,100},
6870 <pmn> \textquotesingle = {300,200},
6871 <ugm> \textquotesingle = {500,500},
6872 <cmr> \textasteriskcentered = {300,200},
6873 <pad> \textasteriskcentered = {500,100},
6874 <pmn> \textasteriskcentered = {200,300},
6875 <ugm> \textasteriskcentered = {300,150},
6876 <pmn> \textfractionsolidus = {-200,-200},
6877 <cmr> \textoneoldstyle = {100, 50},
6878 <pad> \textoneoldstyle = {100, },
6879 <pmn> \textoneoldstyle = { 50, },
6880 <pad> \texttwooldstyle = { 50, },
6881 <pmn> \texttwooldstyle = {-50, },
6882 <cmr> \textthreeoldstyle = {100, 50},
6883 <pmn> \textthreeoldstyle = {-100, },
6884 <cmr> \textfouroldstyle = { 50, 50},
6885 <pad> \textfouroldstyle = { 50,100},
6886 <cmr> \textsevenoldstyle = { 50, 80},
6887 <pad> \textsevenoldstyle = { 50, },
6888 <pmn> \textsevenoldstyle = { 20, },
6889 <cmr> \textlangle = {400, },
6890 <cmr> \textrangle = { ,400},
6891 <cmr|pad> \textminus = {300,300},
6892 <pmn> \textminus = {200,200},
6893 <ugm> \textminus = {250,300},
6894 <pad|pmn> \textlbrackdbl = {100, },
6895 <pad|pmn> \textrbrackdbl = { ,100},
6896 <pmn> \textasciigrave = {300,300},
6897 <cmr|pad|pmn> \texttildelow = {200,250},
6898 <pmn> \textasciibreve = {300,300},
6899 <pmn> \textasciicaron = {300,300},
6900 <pmn> \textacutedbl = {200,300},
6901 <pmn> \textgravedbl = {150,300},
6902 <cmr> \textdagger = {100,100},
6903 <pad> \textdagger = {200,100},
6904 <pmn> \textdagger = { 80, 50},
6905 <ugm> \textdagger = { 80, 80},
6906 <cmr|pad> \textdaggerdbl = { 80, 80},
6907 <pmn> \textdaggerdbl = { 80, 50},
6908 <ugm> \textbardbl = {150,150},
6909 <cmr> \textbullet = {200,100},
6910 <pad> \textbullet = {300, },
6911 <pmn> \textbullet = { 30, 70},
6912 <ugm> \textbullet = { 50,100},
6913 <cmr> \textcelsius = {100, },
6914 <pad> \textcelsius = {200, },
6915 <pmn> \textcelsius = { 50,-50},
6916 <pad> \textflorin = {100, },
6917 <pmn> \textflorin = { 50,100},
6918 <ugm> \textflorin = { ,100},
6919 <cmr> \textcolonmonetary = {150, },
6920 <pad> \textcolonmonetary = {100, },
6921 <pmn> \textcolonmonetary = { 50,-50},
6922 <cmr|pad> \texttrademark = {200, },
6923 <pmn> \texttrademark = { 50,100},
6924 <ugm> \texttrademark = {150, 50},
6925 <ugm> \textcent = { 50, },
6926 <ugm> \textsterling = { , 50},
6927 <ugm> \textbrokenbar = {200,300},
6928 <pmn> \textasciidieresis = {300,200},
6929 <cmr> \textcopyright = {100, },
6930 <pad> \textcopyright = {200,100},
6931 <pmn> \textcopyright = {100,150},
6932 <ugm> \textcopyright = {300, },

```

```

6933 <cmr> \textordfeminine = {100,100},
6934 <pmn> \textordfeminine = {200,200},
6935 <ugm> \textordfeminine = {100,200},
6936 <cmr|pad> \textlnot = {300, },
6937 <pmn|ugm> \textlnot = {200, },
6938 <cmr> \textregistered = {100, },
6939 <pad> \textregistered = {200,100},
6940 <pmn> \textregistered = { 50,150},
6941 <ugm> \textregistered = {300, },
6942 <pmn> \textasciimacron = {150,200},
6943 <cmr|pad> \textdegree = {500,100},
6944 <pmn> \textdegree = {150,150},
6945 <ugm> \textdegree = {300,200},
6946 <cmr> \textpm = {150,100},
6947 <pad> \textpm = {200,150},
6948 <pmn|ugm> \textpm = {150,200},
6949 <cmr> \textonesuperior = {400, },
6950 <pad> \textonesuperior = {300,100},
6951 <pmn> \textonesuperior = {200,100},
6952 <ugm> \textonesuperior = {300,300},
6953 <cmr> \texttwosuperior = {400, },
6954 <pad> \texttwosuperior = {300, },
6955 <pmn> \texttwosuperior = {200,100},
6956 <ugm> \texttwosuperior = {300,200},
6957 <cmr> \textthreesuperior = {400, },
6958 <pad> \textthreesuperior = {300, },
6959 <pmn> \textthreesuperior = {200,100},
6960 <ugm> \textthreesuperior = {300,200},
6961 <ugm> \textmu = { ,100},
6962 <pmn> \textasciiacute = {300,200},
6963 <cmr> \textparagraph = {200, },
6964 <pmn> \textparagraph = { ,100},
6965 <cmr> \textperiodcentered = {500,500},
6966 <pad|pmn|ugm> \textperiodcentered = {300,400},
6967 <cmr> \textordmasculine = {100,100},
6968 <pmn> \textordmasculine = {200,200},
6969 <ugm> \textordmasculine = {300,200},
6970 <cmr> \texteuro = {200, },
6971 <pad> \texteuro = {100, },
6972 <pmn> \texteuro = {100,-50},
6973 <cmr> \texttimes = {200,200},
6974 <pad> \texttimes = {200,100},
6975 <pmn> \texttimes = { 70,100},
6976 <ugm> \texttimes = {200,300},
6977 <cmr|pad> \textdiv = {200,200}
6978 <pmn> \textdiv = {150,200}
6979 <ugm> \textdiv = {200,300},
6980 <ugm> \textsection = { ,200},
6981 <ugm> \textonehalf = { 50,100},
6982 <ugm> \textonequarter = { 50,100},
6983 <ugm> \textthreequarters = { 50,100},
6984 <ugm> \textsurd = { ,100}
6985 }
6986
6987 </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}
```

```
6988 *cmr
6989 \SetProtrusion
6990 [ name = cmr-math-letters ]
6991 { encoding = OML,
6992   family = cmm,
6993   series = {m,b},
6994   shape = it }
6995 {
6996   A = {100, 50}, % \mathnormal
6997   B = { 50,  },
6998   C = { 50,  },
6999   D = { 50, 50},
7000   E = { 50,  },
7001   F = {100, 50},
7002   G = { 50, 50},
7003   H = { 50, 50},
7004   I = { 50, 50},
7005   J = {150, 50},
7006   K = { 50,100},
7007   L = { 50, 50},
7008   M = { 50,  },
7009   N = { 50,  },
7010   O = { 50,  },
7011   P = { 50,  },
7012   Q = { 50, 50},
7013   R = { 50,  },
7014   S = { 50,  },
7015   T = { 50,100},
7016   U = { 50, 50},
7017   V = {100,100},
7018   W = { 50,100},
7019   X = { 50,100},
7020   Y = {100,100},
7021   f = {100,100},
7022   h = {  ,100},
7023   i = {  , 50},
7024   j = {  , 50},
7025   k = {  , 50},
7026   r = {  , 50},
7027   v = {  , 50},
7028   w = {  , 50},
7029   x = {  , 50},
7030   "0B = { 50,100}, % \alpha
7031   "0C = { 50, 50}, % \beta
7032   "0D = {200,150}, % \gamma
7033   "0E = { 50, 50}, % \delta
7034   "0F = { 50, 50}, % \epsilon
7035   "10 = { 50,150}, % \zeta
7036   "12 = { 50,  }, % \theta
7037   "13 = {  ,100}, % \iota
7038   "14 = {  ,100}, % \kappa
7039   "15 = {100, 50}, % \lambda
7040   "16 = {  , 50}, % \mu
7041   "17 = {  , 50}, % \nu
```

```

7042 "18 = { , 50}, % \xi
7043 "19 = { 50,100}, % \pi
7044 "1A = { 50, 50}, % \rho
7045 "1B = { ,150}, % \sigma
7046 "1C = { 50,150}, % \tau
7047 "1D = { 50, 50}, % \upsilon
7048 "1F = { 50,100}, % \chi
7049 "20 = { 50, 50}, % \psi
7050 "21 = { , 50}, % \omega
7051 "22 = { , 50}, % \varepsilon
7052 "23 = { , 50}, % \vartheta
7053 "24 = { , 50}, % \varpi
7054 "25 = {100, }, % \varrho
7055 "26 = {100,100}, % \varsigma
7056 "27 = { 50, 50}, % \varphi
7057 "28 = {100,100}, % \leftharpoonup
7058 "29 = {100,100}, % \leftharpoondown
7059 "2A = {100,100}, % \rightharpoonup
7060 "2B = {100,100}, % \rightharpoondown
7061 "2C = {300,200}, % \lhook
7062 "2D = {200,300}, % \rhook
7063 "2E = { ,100}, % \triangleright
7064 "2F = {100, }, % \triangleleft
7065 "3A = { ,500}, % ., \ldotp
7066 "3B = { ,500}, % ,
7067 "3C = {200,100}, % <
7068 "3D = {300,400}, % /
7069 "3E = {100,200}, % >
7070 "3F = {200,200}, % \star
7071 "5B = { ,100}, % \flat
7072 "5E = {200,200}, % \smile
7073 "5F = {200,200}, % \frown
7074 "7C = {100, }, % \jmath
7075 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

7076 }
7077

```

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}

```

```

7078 \SetProtrusion
7079 [ name = cmr-math-symbols ]
7080 { encoding = OMS,
7081   family = cmsy,
7082   series = {m,b},
7083   shape = n }
7084 {
7085   A = {150, 50}, % \mathcal
7086   C = { ,100},
7087   D = { , 50},
7088   F = { 50,150},
7089   I = { ,100},
7090   J = {100,150},
7091   K = { ,100},
7092   L = {100, },
7093   M = { 50, 50},
7094   N = { 50,100},
7095   P = { , 50},
7096   Q = { 50, },
7097   R = { , 50},
7098   T = { 50,150},
7099   V = { 50, 50},

```

```

7100     W = {    , 50},
7101     X = {100,100},
7102     Y = {100,   },
7103     Z = {100,150},
7104     "00 = {300,300}, % -
7105     "01 = {    ,700}, % \cdot, \cdotp
7106     "02 = {150,250}, % \times
7107     "03 = {150,250}, % *, \ast
7108     "04 = {200,300}, % \div
7109     "05 = {150,250}, % \diamond
7110     "06 = {200,200}, % \pm
7111     "07 = {200,200}, % \mp
7112     "08 = {100,100}, % \oplus
7113     "09 = {100,100}, % \ominus
7114     "0A = {100,100}, % \otimes
7115     "0B = {100,100}, % \oslash
7116     "0C = {100,100}, % \odot
7117     "0D = {100,100}, % \bigcirc
7118     "0E = {100,100}, % \circ
7119     "0F = {100,100}, % \bullet
7120     "10 = {100,100}, % \asymp
7121     "11 = {100,100}, % \equiv
7122     "12 = {200,100}, % \subseteq
7123     "13 = {100,200}, % \supseteq
7124     "14 = {200,100}, % \leq
7125     "15 = {100,200}, % \geq
7126     "16 = {200,100}, % \preceq
7127     "17 = {100,200}, % \succeq
7128     "18 = {200,200}, % \sim
7129     "19 = {150,150}, % \approx
7130     "1A = {200,100}, % \subset
7131     "1B = {100,200}, % \supset
7132     "1C = {200,100}, % \ll
7133     "1D = {100,200}, % \gg
7134     "1E = {300,100}, % \prec
7135     "1F = {100,300}, % \succ
7136     "20 = {100,200}, % \leftarrow
7137     "21 = {200,100}, % \rightarrow
7138     "22 = {100,100}, % \uparrow
7139     "23 = {100,100}, % \downarrow
7140     "24 = {100,100}, % \leftrightarrows
7141     "25 = {100,100}, % \nearrow
7142     "26 = {100,100}, % \searrow
7143     "27 = {100,100}, % \simeq
7144     "28 = {100,100}, % \Leftarrow
7145     "29 = {100,100}, % \Rightarrow
7146     "2A = {100,100}, % \Uparrow
7147     "2B = {100,100}, % \Downarrow
7148     "2C = {100,100}, % \Leftrightarrow
7149     "2D = {100,100}, % \nrightarrow
7150     "2E = {100,100}, % \swarrow
7151     "2F = {    ,100}, % \propto
7152     "30 = {    ,400}, % \prime
7153     "31 = {100,100}, % \infty
7154     "32 = {150,100}, % \in
7155     "33 = {100,150}, % \ni
7156     "34 = {100,100}, % \triangle, \bigtriangleup
7157     "35 = {100,100}, % \bigtriangledown
7158     "38 = {    ,100}, % \forall
7159     "39 = {100,   }, % \exists
7160     "3A = {200,   }, % \neg
7161     "3E = {200,200}, % \top
7162     "3F = {200,200}, % \bot, \perp
7163     "5E = {100,200}, % \wedge
7164     "5F = {100,200}, % \vee

```

```

7165 "60 = { ,300}, % \vdash
7166 "61 = {300, }, % \dashv
7167 "62 = {100,100}, % \lfloor
7168 "63 = {100,100}, % \rfloor
7169 "64 = {100,100}, % \lceil
7170 "65 = {100,100}, % \rceil
7171 "66 = {150, }, % \lbrace
7172 "67 = { ,150}, % \rbrace
7173 "68 = {400, }, % \langle
7174 "69 = { ,400}, % \rangle
7175 "6C = {100,100}, % \updownarrow
7176 "6D = {100,100}, % \Updownarrow
7177 "6E = {100,300}, % \, \backslash, \setminus
7178 "72 = {100,100}, % \nabla
7179 "79 = {200,200}, % \dagger
7180 "7A = {100,100}, % \ddagger
7181 "7B = {100, }, % \mathparagraph
7182 "7C = {100,100}, % \clubsuit
7183 "7D = {100,100}, % \diamondsuit
7184 "7E = {100,100}, % \heartsuit
7185 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

7186 }
7187

```

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}{cmx}{m}{n}
```

```

7188 </cmr>
7189 </cfg-t>

```

### 15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
7190 <*cfg-u>
```

Symbol font 'a'.

```

7191 <*msa>
7192 \SetProtrusion
7193 [ name = AMS-a ]
7194 { encoding = U,
7195   family = msa }
7196 {
7197 "05 = {150,250}, % \centerdot
7198 "06 = {100,100}, % \lozenge
7199 "07 = { 50, 50}, % \blacklozenge
7200 "08 = { 50, 50}, % \circlearrowright
7201 "09 = { 50, 50}, % \circlearrowleft
7202 "0A = {100,100}, % \rightleftharpoons
7203 "0B = {100,100}, % \leftrightharpoons
7204 "0D = {-50,200}, % \Vdash
7205 "0E = {-50,200}, % \Vvdash
7206 "0F = {-70,150}, % \vDash
7207 "10 = {100,150}, % \twoheadrightarrow
7208 "11 = {100,150}, % \twoheadleftarrow
7209 "12 = { 50,100}, % \leftleftarrows
7210 "13 = { 50, 80}, % \rightrightarrows
7211 "14 = {120,120}, % \upuparrows
7212 "15 = {120,120}, % \downdownarrows
7213 "16 = {200,200}, % \upharpoonright
7214 "17 = {200,200}, % \downharpoonright

```

```

7215 "18 = {200,200}, % \upharpoonleft
7216 "19 = {200,200}, % \downharpoonleft
7217 "1A = { 80,100}, % \rightarrowtail
7218 "1B = { 80,100}, % \leftarrowtail
7219 "1C = { 50, 50}, % \leftrightarrows
7220 "1D = { 50, 50}, % \rightleftarrows
7221 "1E = {250,  }, % \Lsh
7222 "1F = {  ,250}, % \Rsh
7223 "20 = {100,100}, % \rightsquigarrow
7224 "21 = {100,100}, % \leftrightsquigarrow
7225 "22 = {100, 50}, % \looparrowleft
7226 "23 = { 50,100}, % \looparrowright
7227 "24 = { 50, 80}, % \circeq
7228 "25 = {  ,100}, % \succsim
7229 "26 = {  ,100}, % \gtrsim
7230 "27 = {  ,100}, % \gtrapprox
7231 "28 = {150, 50}, % \multimap
7232 "2B = {100,150}, % \doteqdot
7233 "2C = {100,150}, % \triangleq
7234 "2D = {100, 50}, % \precsim
7235 "2E = {100, 50}, % \lessim
7236 "2F = { 50, 50}, % \lessapprox
7237 "30 = {100, 50}, % \eqslantless
7238 "31 = { 50, 50}, % \eqslantgtr
7239 "32 = {100, 50}, % \curlyeqprec
7240 "33 = { 50,100}, % \curlyeqsucc
7241 "34 = {100, 50}, % \preccurlyeq
7242 "36 = { 50,  }, % \leqslant
7243 "38 = {  , 50}, % \backprime
7244 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
7245 "3C = { 50,100}, % \succcurlyeq
7246 "3E = {  , 50}, % \geqslant
7247 "40 = {  , 50}, % \sqsubset
7248 "41 = { 50,  }, % \sqsupset
7249 "42 = {  ,150}, % \vartriangleright, \rhd
7250 "43 = {150,  }, % \vartriangleleft, \lhd
7251 "44 = {  ,100}, % \trianglerighteq, \unrhd
7252 "45 = {100,  }, % \trianglelefteq, \unlhd
7253 "46 = {100,100}, % \bigstar
7254 "48 = { 50, 50}, % \blacktriangledown
7255 "49 = {  ,100}, % \blacktriangleright
7256 "4A = {100,  }, % \blacktriangleleft
7257 "4B = {  ,150}, % \dashrightarrow (the arrow)
7258 "4C = {150,  }, % \dashleftarrow
7259 "4D = { 50, 50}, % \vartriangle
7260 "4E = { 50, 50}, % \blacktriangle
7261 "4F = { 50, 50}, % \triangledown
7262 "50 = { 50, 50}, % \eqcirc
7263 "56 = {  ,150}, % \Rrightarrow
7264 "57 = {150,  }, % \Lleftarrow
7265 "58 = {100,300}, % \checkmark
7266 "5C = { 50, 50}, % \angle
7267 "5D = { 50, 50}, % \measuredangle
7268 "5E = { 50, 50}, % \sphericalangle
7269 "5F = {  , 50}, % \varpropto
7270 "60 = {100,100}, % \smallsmile
7271 "61 = {100,100}, % \smallfrown
7272 "62 = { 50,  }, % \Subset
7273 "63 = {  , 50}, % \Supset
7274 "66 = {150,150}, % \curlywedge
7275 "67 = {150,150}, % \curlyvee
7276 "68 = { 50,150}, % \leftthreetimes
7277 "69 = {100, 50}, % \rightthreetimes
7278 "6C = { 50, 50}, % \bumpeq
7279 "6D = { 50, 50}, % \Bumpeq

```

```

7280 "6E = {100, }, % \l11
7281 "6F = { ,100}, % \ggg
7282 "70 = { 50,100}, % \ulcorner
7283 "71 = {100, 50}, % \urcorner
7284 "75 = {150,200}, % \dotplus
7285 "76 = { 50,100}, % \backsim
7286 "78 = { 50,100}, % \llcorner
7287 "79 = {100, 50}, % \lrcorner
7288 "7C = {100,100}, % \intercal
7289 "7D = { 50, 50}, % \circledcirc
7290 "7E = { 50, 50}, % \circledast
7291 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

7292 }
7293
7294 </msa>

```

Symbol font 'b'.

```

7295 <*msb>
7296 \SetProtrusion
7297 [ name = AMS-b ]
7298 { encoding = U,
7299   family = msb }
7300 {
7301   A = { 50, 50}, % \mathbb
7302   C = { 50, 50},
7303   G = { , 50},
7304   L = { , 50},
7305   P = { , 50},
7306   R = { , 50},
7307   T = { , 50},
7308   V = { 50, 50},
7309   X = { 50, 50},
7310   Y = { 50, 50},
7311 "00 = { 50, 50}, % \lvertneqq
7312 "01 = { 50, 50}, % \gvertneqq
7313 "02 = { 50, 50}, % \nleq
7314 "03 = { 50, 50}, % \ngeq
7315 "04 = {100, 50}, % \nless
7316 "05 = { 50,150}, % \ngtr
7317 "06 = {100, 50}, % \nprec
7318 "07 = { 50,150}, % \nsucc
7319 "08 = { 50, 50}, % \lneqq
7320 "09 = { 50, 50}, % \gneqq
7321 "0A = {100,100}, % \leqslant
7322 "0B = {100,100}, % \geqslant
7323 "0C = {100, 50}, % \lneq
7324 "0D = { 50,100}, % \gneq
7325 "0E = {100, 50}, % \npreceq
7326 "0F = { 50,100}, % \nsucceq
7327 "10 = { 50, }, % \precnsim
7328 "11 = { 50, 50}, % \succnsim
7329 "12 = { 50, 50}, % \lnsim
7330 "13 = { 50, 50}, % \gnsim
7331 "14 = { 50, 50}, % \leqq
7332 "15 = { 50, 50}, % \geqq
7333 "16 = { 50, 50}, % \precneqq
7334 "17 = { 50, 50}, % \succneqq
7335 "18 = { 50, 50}, % \precnapprox
7336 "19 = { 50, 50}, % \succnapprox
7337 "1A = { 50, 50}, % \lnapprox
7338 "1B = { 50, 50}, % \gnapprox
7339 "1C = {150,200}, % \nsim
7340 "1D = { 50, 50}, % \ncong

```

```

7341 "1E = {100,150}, % \diagup
7342 "1F = {100,150}, % \diagdown
7343 "20 = {100, 50}, % \varsubsetneq
7344 "21 = { 50,100}, % \varsupsetneq
7345 "22 = {100, 50}, % \subsetneqq
7346 "23 = { 50,100}, % \supsetneqq
7347 "24 = {100, 50}, % \subsetneqq
7348 "25 = { 50,100}, % \supsetneqq
7349 "26 = {100, 50}, % \varsubsetneqq
7350 "27 = { 50,100}, % \varsupsetneqq
7351 "28 = {100, 50}, % \subsetneq
7352 "29 = { 50,100}, % \supsetneq
7353 "2A = {100, 50}, % \subseteq
7354 "2B = { 50,100}, % \supseteq
7355 "2C = { 50,100}, % \nparallel
7356 "2D = {100,150}, % \nmid
7357 "2E = {150,150}, % \shortmid
7358 "2F = {100,100}, % \shortparallel
7359 "30 = { ,150}, % \nvDash
7360 "31 = { ,150}, % \nVDash
7361 "32 = { ,100}, % \nvDash
7362 "33 = { ,100}, % \nVDash
7363 "34 = { ,100}, % \ntrianglerighteq
7364 "35 = {100, }, % \trianglelefteq
7365 "36 = {100, }, % \triangleleft
7366 "37 = { ,100}, % \triangleright
7367 "38 = {100,200}, % \leftarrow
7368 "39 = {100,200}, % \rightarrow
7369 "3A = {100,100}, % \Leftarrow
7370 "3B = { 50,100}, % \Rightarrow
7371 "3C = {100,100}, % \Leftrightarrow
7372 "3D = {100,200}, % \leftrightarrows
7373 "3E = { 50, 50}, % \divideontimes
7374 "3F = { 50, 50}, % \varepsilon
7375 "60 = {200, }, % \Finv
7376 "61 = { , 50}, % \Game
7377 "68 = {100,100}, % \eqsim
7378 "69 = { 50, }, % \beth
7379 "6A = { 50, }, % \gimel
7380 "6B = {150, }, % \daleth
7381 "6C = {200, }, % \lessdot
7382 "6D = { ,200}, % \gtrdot
7383 "6E = {100,200}, % \ltimes
7384 "6F = {150,100}, % \rtimes
7385 "70 = { 50,100}, % \shortmid
7386 "71 = { 50, 50}, % \shortparallel
7387 "72 = {200,300}, % \smallsetminus
7388 "73 = {100,200}, % \thicksim
7389 "74 = { 50,100}, % \thickapprox
7390 "75 = { 50, 50}, % \approx
7391 "76 = { 50,100}, % \succapprox
7392 "77 = { 50, 50}, % \precapprox
7393 "78 = {100,100}, % \curvearrowleft
7394 "79 = { 50,150}, % \curvearrowright
7395 "7A = { 50,200}, % \digamma
7396 "7B = {100, 50}, % \varkappa
7397 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

7398 }
7399
7400 (/msb)

```

**15.8.8 Euler**

Euler Roman font (package euler).

```

7401 (*eur)
7402 \SetProtrusion
7403 [ name = euler ]
7404 { encoding = U,
7405   family = eur }
7406 {
7407   "01 = {100,100},
7408   "03 = {100,150},
7409   "06 = { ,100},
7410   "07 = {100,150},
7411   "08 = {100,100},
7412   "0A = {100,100},
7413   "0B = { , 50},
7414   "0C = { ,100},
7415   "0D = {100,100},
7416   "0E = { ,100},
7417   "0F = {100,100},
7418   "10 = {100,100},
7419   "13 = { ,100},
7420   "14 = { ,100},
7421   "15 = { , 50},
7422   "16 = { , 50},
7423   "17 = { 50,100},
7424   "18 = { 50,100},
7425   "1A = { , 50},
7426   "1B = { , 50},
7427   "1C = { 50,100},
7428   "1D = { 50,100},
7429   "1E = { 50,100},
7430   "1F = { 50,100},
7431   "20 = { , 50},
7432   "21 = { , 50},
7433   "22 = { 50,100},
7434   "24 = { , 50},
7435   "27 = { 50,100},
7436   1 = {100,100},
7437   7 = { 50,100},
7438   "3A = {300,500},
7439   "3B = {200,400},
7440   "3C = {200,100},
7441   "3D = {200,200},
7442   "3E = {100,200},
7443   A = { ,100},
7444   D = { , 50},
7445   J = { 50, },
7446   K = { , 50},
7447   L = { , 50},
7448   Q = { , 50},
7449   T = { 50, },
7450   X = { 50, 50},
7451   Y = { 50, },
7452   h = { , 50},
7453   k = { , 50}
7454 }
7455

```

Extended by the eulervm package.

```

7456 \SetProtrusion
7457 [ name = euler-vm,
7458   load = euler ]
7459 { encoding = U,
7460   family = zeur }

```

```

7461 {
7462 "28 = {100,200},
7463 "29 = {100,200},
7464 "2A = {100,150},
7465 "2B = {100,150},
7466 "2C = {200,300},
7467 "2D = {200,300},
7468 "2E = { ,100},
7469 "2F = {100, },
7470 "3F = {150,150},
7471 "5B = { ,100},
7472 "5E = {100,100},
7473 "5F = {100,100},
7474 "80 = { , 50},
7475 "81 = {200,250},
7476 "82 = {100,200}
7477 }
7478
7479 (/eur)

```

### Euler Script font (euca1).

```

7480 (*eus)
7481 \SetProtrusion
7482 [ name = euscript ]
7483 { encoding = U,
7484 family = eus }
7485 {
7486 A = {100,100},
7487 B = { 50,100},
7488 C = { 50, 50},
7489 D = { 50,100},
7490 E = { 50,100},
7491 F = { 50, },
7492 G = { 50, },
7493 H = { ,100},
7494 K = { , 50},
7495 L = { ,150},
7496 M = { , 50},
7497 N = { , 50},
7498 O = { 50, 50},
7499 P = { 50, 50},
7500 T = { ,100},
7501 U = { , 50},
7502 V = { 50, 50},
7503 W = { 50, 50},
7504 X = { 50, 50},
7505 Y = { 50, },
7506 Z = { 50,100},
7507 "00 = {250,250},
7508 "18 = {200,200},
7509 "3A = {200,150},
7510 "40 = { ,100},
7511 "5E = {100,100},
7512 "5F = {100,100},
7513 "66 = { 50, },
7514 "67 = { , 50},
7515 "6E = {200,200}
7516 }
7517
7518 \SetProtrusion
7519 [ name = euscript-vm,
7520 load = euscript ]
7521 { encoding = U,
7522 family = zeus }
7523 {

```

```
7524 "01 = {600,600},
7525 "02 = {200,200},
7526 "03 = {200,200},
7527 "04 = {200,200},
7528 "05 = {150,150},
7529 "06 = {200,200},
7530 "07 = {200,200},
7531 "08 = {100,100},
7532 "09 = {100,100},
7533 "0A = {100,100},
7534 "0B = {100,100},
7535 "0C = {100,100},
7536 "0D = {100,100},
7537 "0E = {150,150},
7538 "0F = {100,100},
7539 "10 = {150,150},
7540 "11 = {100,100},
7541 "12 = {150,100},
7542 "13 = {100,150},
7543 "14 = {150,100},
7544 "15 = {100,150},
7545 "16 = {200,100},
7546 "17 = {100,200},
7547 "19 = {150,150},
7548 "1A = {150,100},
7549 "1B = {100,150},
7550 "1C = {100,100},
7551 "1D = {100,100},
7552 "1E = {250,100},
7553 "1F = {100,250},
7554 "20 = {150,200},
7555 "21 = {150,200},
7556 "22 = {150,150},
7557 "23 = {150,150},
7558 "24 = {100,200},
7559 "25 = {150,150},
7560 "26 = {150,150},
7561 "27 = {100,100},
7562 "28 = {100,100},
7563 "29 = {100,150},
7564 "2A = {100,100},
7565 "2B = {100,100},
7566 "2C = {100,100},
7567 "2D = {150,150},
7568 "2E = {150,150},
7569 "2F = {100,100},
7570 "30 = {100,100},
7571 "31 = {100,100},
7572 "32 = {100,100},
7573 "33 = {100,100},
7574 "34 = {100,100},
7575 "35 = {100,100},
7576 "3E = {150,150},
7577 "3F = {150,150},
7578 "60 = { ,200},
7579 "61 = {200, },
7580 "62 = {100,100},
7581 "63 = {100,100},
7582 "64 = {100,100},
7583 "65 = {100,100},
7584 "68 = {300, },
7585 "69 = { ,300},
7586 "6C = {100,100},
7587 "6D = {100,100},
7588 "6F = {100,100},
```

```

7589     "72 = {100,100},
7590     "73 = {200,100},
7591     "76 = {   ,100},
7592     "77 = {100,   },
7593     "78 = { 50, 50},
7594     "79 = {100,100},
7595     "7A = {100,100},
7596     "7D = {150,150},
7597     "7E = {100,100},
7598     "A8 = {100,100},
7599     "A9 = {100,100},
7600     "AB = {200,200},
7601     "BA = {   ,200},
7602     "BB = {   ,200},
7603     "BD = {200,200},
7604     "DE = {200,200}
7605   }
7606
7607 (/eus)

```

### Euler Fraktur font (eufrak).

```

7608 (*euf)
7609 \SetProtrusion
7610   [ name      = mathfrak ]
7611   { encoding = U,
7612     family   = euf   }
7613   {
7614     A = {   , 50},
7615     B = {   , 50},
7616     C = { 50, 50},
7617     D = {   , 80},
7618     E = { 50,   },
7619     G = {   , 50},
7620     L = {   , 80},
7621     O = {   , 50},
7622     T = {   , 80},
7623     X = { 80, 50},
7624     Z = { 80, 50},
7625     b = {   , 50},
7626     c = {   , 50},
7627     k = {   , 50},
7628     p = {   , 50},
7629     q = { 50,   },
7630     v = {   , 50},
7631     w = {   , 50},
7632     x = {   , 50},
7633     1 = {100,100},
7634     2 = { 80, 80},
7635     3 = { 80, 50},
7636     4 = { 80, 50},
7637     7 = { 50, 50},
7638     "12 = {500,500},
7639     "13 = {500,500},
7640     ! = {   ,200},
7641     ' = {200,300},
7642     ( = {200,   },
7643     ) = {   ,200},
7644     * = {200,200},
7645     + = {200,250},
7646     - = {200,200},
7647     {,} = {300,300},
7648     . = {400,400},
7649     {=} = {200,200},
7650     : = {   ,200},
7651     ; = {   ,200},

```

```

7652     ] = { ,200}
7653   }
7654
7655 </euf>
7656 </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>).

```

7657 <*cfg-e>
7658 \SetProtrusion
7659 <zpeu|euroitc> { encoding = U,
7660 <mvs> { encoding = {OT1,U},
7661 <zpeu> family = zpeu }
7662 <euroitc> family = {euroitc,euroitcs }
7663 <mvs> family = mvs }
7664 {
7665 <zpeu> E = {50, }
7666 <euroitc> E = {100,50}
7667 <mvs> 164 = {50,50}, % \EUR
7668 <mvs> 068 = {50,-100} % \EURdig
7669 }
7670
7671 <*zpeu|euroitc>
7672 \SetProtrusion
7673 { encoding = U,
7674 <zpeu> family = zpeu,
7675 <euroitc> family = {euroitc,euroitcs},
7676 shape = it* }
7677 {
7678 <zpeu> E = {100,-50}
7679 <euroitc> E = {100,}
7680 }
7681
7682 </zpeu|euroitc>
7683 <*zpeu>
7684 \SetProtrusion
7685 { encoding = U,
7686 family = {zpeus,eurosans} }
7687 {
7688 E = {100,50}
7689 }
7690
7691 \SetProtrusion
7692 { encoding = U,
7693 family = {zpeus,eurosans},
7694 shape = it* }
7695 {
7696 E = {200, }
7697 }
7698
7699 </zpeu>
7700 </cfg-e>

```

## 15.9 Interword spacing

Default unit is space.

```

7701 <*m-t|cmr>
7702 %%% -----

```

---

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

```

7703 %% INTERWORD SPACING
7704
7705 </m-t|cmr>
7706 <+m-t>
7707 \SetExtraSpacing
7708   [ name = default ]
7709   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
7710   {

```

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

```
7711   { , } = { , -500, 500 },
```

- in front of capitals which have optical more room on their left side, e.g., ‘A’, ‘J’, ‘T’, ‘V’, ‘W’, and ‘Y’ [this is not yet possible – RS]
- in front of capitals which have circle/oval shapes on their left side, e.g., ‘C’, ‘G’, ‘O’, and ‘Q’ [ditto – RS]
- after ‘r’ (because of the bigger optical room on the righthand side)

```
7712   r = { , -300, 300 },
```

- [before or] after lowercase characters with ascenders

```
7713   b = { , -200, 200 },
```

```
7714   d = { , -200, 200 },
```

```
7715   f = { , -200, 200 },
```

```
7716   h = { , -200, 200 },
```

```
7717   k = { , -200, 200 },
```

```
7718   l = { , -200, 200 },
```

```
7719   t = { , -200, 200 },
```

- [before or] after lowercase characters with x-height plus descender with additional optical space, e.g., ‘v’, or ‘w’

```
7720   c = { , -100, 100 },
```

```
7721   p = { , -100, 100 },
```

```
7722   v = { , -100, 100 },
```

```
7723   w = { , -100, 100 },
```

```
7724   z = { , -100, 100 },
```

```
7725   x = { , -100, 100 },
```

```
7726   y = { , -100, 100 },
```

- [before or] after lowercase characters with x-height plus descender without additional optical space

```
7727     i = { , 50, -50},
7728     m = { , 50, -50},
7729     n = { , 50, -50},
7730     u = { , 50, -50},
```

- after colon and semicolon

```
7731     : = { ,200,-200},
7732     ; = { ,200,-200},
```

- after punctuation which ends a sentence, e.g., period, exclamation mark, question mark

```
7733     . = { ,250,-250},
7734     ! = { ,250,-250},
7735     ? = { ,250,-250}
```

The order has to be reversed when enlarging is needed.’

```
7736     }
7737
7738 </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>

```
7739 <+cmr>
7740 \SetExtraSpacing
7741 [ name = T2A,
7742   load = default ]
7743 { encoding = T2A,
7744   family = cmr }
7745 {
7746   \cyrg = { ,-300,300},
7747   \cyrb = { ,-200,200},
7748   \cyrk = { ,-200,200},
7749   \cyrs = { ,-100,100},
7750   \cyrr = { ,-100,100},
7751   \cyrh = { ,-100,100},
7752   \cyru = { ,-100,100},
7753   \cyrt = { , 50, -50},
7754   \cyrp = { , 50, -50},
7755   \cyri = { , 50, -50},
```

```

7756     \cyrishrt = { , 50, -50},
7757   }
7758

```

### 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.

```

7759 \SetExtraSpacing
7760   [ name = nonfrench-cmr,
7761     load = default,
7762     context = nonfrench ]
7763   { encoding = {OT1,T1,LY1,OT4,QX,T5},
7764     family = cmr }
7765   {

```

`latex.ltx` has:

```

\def\nonfrenchspacing{
  \sfcode`\ . 3000
  \sfcode`\? 3000
  \sfcode`\! 3000

```

```

7766   . = {333,2000,-667},
7767   ? = {333,2000,-667},
7768   ! = {333,2000,-667},

```

```

\sfcodes\ : 2000

```

```

7769   : = {333,1000,-500},

```

```

\sfcodes\ ; 1500

```

```

7770   ; = { , 500,-333},

```

```

\sfcodes\ , 1250

```

```

7771   {,}= { , 250,-200}

```

```

}

```

```

7772   }
7773
7774 </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.

```

7775 < *m-t >
7776 \SetExtraSpacing

```

```

7777 [ name      = nonfrench-default,
7778   load      = default,
7779   context   = nonfrench ]
7780 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
7781 {
7782   . = {240,2000,-667},
7783   ? = {240,2000,-667},
7784   ! = {240,2000,-667},
7785   : = {240,1000,-500},
7786   ; = {   , 500,-333},
7787   {,}= {   , 250,-200}
7788 }
7789

```

## 15.10 Additional kerning

Default unit is 1 em.

```

7790 %%% -----
7791 %%% ADDITIONAL KERNING
7792

```

A dummy list to be loaded when no context is active.

```

7793 \SetExtraKerning
7794 [ name = empty ]
7795 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
7796 { }
7797

```

### 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.)

```

7798 \SetExtraKerning
7799 [ name      = french-default,
7800   context   = french,
7801   unit      = space ]
7802 { encoding = {OT1,T1,LY1} }
7803 {
7804   : = {1000,}, % = \fontdimen2
7805   ; = {500, }, % ~ \thinspace
7806   ! = {500, },
7807   ? = {500, }
7808 }
7809

```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTeX.

```

7810 \SetExtraKerning
7811 [ name      = french-guillemets,
7812   context   = french-guillemets,
7813   load      = french-default,
7814   unit      = space ]

```

---

25 [http://fr.wikipedia.org/wiki/Espace\\_typographique](http://fr.wikipedia.org/wiki/Espace_typographique), 5 July 2007.

```
7815 { encoding = {T1,LY1} }
7816 {
7817 \guillemotleft = { ,800}, % = 0.8\fontdimen2
7818 \guillemotright = {800, }
7819 }
7820
7821 \SetExtraKerning
7822 [ name = french-guillemets-OT1,
7823 context = french-guillemets,
7824 load = french-default,
7825 unit = space ]
7826 { encoding = OT1 }
7827 { }
7828
```

### 15.10.2 Turkish

```
7829 \SetExtraKerning
7830 [ name = turkish,
7831 context = turkish ]
7832 { encoding = {OT1,T1,LY1} }
7833 {
7834 : = {167, }, % = \thinspace
7835 ! = {167, },
7836 {=} = {167, }
7837 }
7838
7839 /m-t
7840 /config
```





7946 T = {T, T̄, 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, 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, T̿̿̄, T̿̿̇, T̿̿̈, T̿̿̉, T̿̿̊, T̿̿̋, T̿̿̌, T̿̿̍, T̿̿̎, T̿̿̏, T̿̿̐, T̿̿̑, T̿̿̒, T̿̿̓, T̿̿̔, T̿̿̕, T̖̿̿, T̗̿̿, T̘̿̿, T̙̿̿, T̿̿̚, T̛̿̿, T̜̿̿, T̝̿̿, T̞̿̿, T̟̿̿, T̠̿̿, T̡̿̿, T̢̿̿, Ṭ̿̿, T̤̿̿, T̥̿̿, Ț̿̿, Ţ̿̿, T̨̿̿, T̩̿̿, T̪̿̿, T̫̿̿, T̬̿̿, Ṱ̿̿, T̮̿̿, T̯̿̿, T̰̿̿, Ṯ̿̿, T̲̿̿, T̳̿̿, T̴̿̿, T̵̿̿, T̶̿̿, T̷̿̿, T̸̿̿, T̹̿̿, T̺̿̿, T̻̿̿, T̼̿̿, T̿̿̽, T̿̿̾, T̿̿̿, T̿̿̿̄, T̿̿̿̇, T̿̿̿̈, T̿̿̿̉, T̿̿̿̊, T̿̿̿̋, T̿̿̿̌, T̿̿̿̍, T̿̿̿̎, T̿̿̿̏, T̿̿̿̐, T̿̿̿̑, T̿̿̿̒, T̿̿̿̓, T̿̿̿̔, T̿̿̿̕, T̖̿̿̿, T̗̿̿̿, T̘̿̿̿, T̙̿̿̿, T̿̿̿̚, T̛̿̿̿, T̜̿̿̿, T̝̿̿̿, T̞̿̿̿, T̟̿̿̿, T̠̿̿̿, T̡̿̿̿, T̢̿̿̿, Ṭ̿̿̿, T̤̿̿̿, T̥̿̿̿, Ț̿̿̿, Ţ̿̿̿, T̨̿̿̿, T̩̿̿̿, T̪̿̿̿, T̫̿̿̿, T̬̿̿̿, Ṱ̿̿̿, T̮̿̿̿, T̯̿̿̿, T̰̿̿̿, Ṯ̿̿̿, T̲̿̿̿, T̳̿̿̿, T̴̿̿̿, T̵̿̿̿, T̶̿̿̿, T̷̿̿̿, T̸̿̿̿, T̹̿̿̿, T̺̿̿̿, T̻̿̿̿, T̼̿̿̿, T̿̿̿̽, T̿̿̿̾, T̿̿̿̿}, % Cyr  
7947 T, T̄}, % Cyr  
7948 U = {U, Ū, Ŭ, Ů, Ű, Ų, Ŵ, Ŷ, Ÿ, ź, Ż, ż, Ž, ſ, ſ̄, 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, ſ̿̄, ſ̿̇, ſ̿̈, ſ̿̉, ſ̿̊, ſ̿̋, ſ̿̌, ſ̿̍, ſ̿̎, ſ̿̏, ſ̿̐, ſ̿̑, ſ̿̒, ſ̿̓, ſ̿̔, ſ̿̕, ſ̖̿, ſ̗̿, ſ̘̿, ſ̙̿, ſ̿̚, ſ̛̿, ſ̜̿, ſ̝̿, ſ̞̿, ſ̟̿, ſ̠̿, ſ̡̿, ſ̢̿, ſ̣̿, ſ̤̿, ſ̥̿, ſ̦̿, ſ̧̿, ſ̨̿, ſ̩̿, ſ̪̿, ſ̫̿, ſ̬̿, ſ̭̿, ſ̮̿, ſ̯̿, ſ̰̿, ſ̱̿, ſ̲̿, ſ̳̿, ſ̴̿, ſ̵̿, ſ̶̿, ſ̷̿, ſ̸̿, ſ̹̿, ſ̺̿, ſ̻̿, ſ̼̿, ſ̿̽, ſ̿̾, ſ̿̿, ſ̿̿̄, ſ̿̿̇, ſ̿̿̈, ſ̿̿̉, ſ̿̿̊, ſ̿̿̋, ſ̿̿̌, ſ̿̿̍, ſ̿̿̎, ſ̿̿̏, ſ̿̿̐, ſ̿̿̑, ſ̿̿̒, ſ̿̿̓, ſ̿̿̔, ſ̿̿̕, ſ̖̿̿, ſ̗̿̿, ſ̘̿̿, ſ̙̿̿, ſ̿̿̚, ſ̛̿̿, ſ̜̿̿, ſ̝̿̿, ſ̞̿̿, ſ̟̿̿, ſ̠̿̿, ſ̡̿̿, ſ̢̿̿, ſ̣̿̿, ſ̤̿̿, ſ̥̿̿, ſ̦̿̿, ſ̧̿̿, ſ̨̿̿, ſ̩̿̿, ſ̪̿̿, ſ̫̿̿, ſ̬̿̿, ſ̭̿̿, ſ̮̿̿, ſ̯̿̿, ſ̰̿̿, ſ̱̿̿, ſ̲̿̿, ſ̳̿̿, ſ̴̿̿, ſ̵̿̿, ſ̶̿̿, ſ̷̿̿, ſ̸̿̿, ſ̹̿̿, ſ̺̿̿, ſ̻̿̿, ſ̼̿̿, ſ̿̿̽, ſ̿̿̾, ſ̿̿̿}, % Cyr  
7949 V = {V, V̄},  
7950 W = {W, W̄, Ẇ, Ẅ, W̉, W̊, W̋, W̌, W̍, W̎, W̏, W̐, W̑, W̒, W̓, W̔, W̕, W̖, W̗, W̘, W̙, W̚, W̛, W̜, W̝, W̞, W̟, W̠, W̡, W̢, Ẉ, W̤, W̥, W̦, W̧, W̨, W̩, W̪, W̫, W̬, W̭, W̮, W̯, W̰, W̱, W̲, W̳, W̴, W̵, W̶, W̷, W̸, W̹, W̺, W̻, W̼, W̽, W̾, W̿, W̿̄, W̿̇, W̿̈, W̿̉, W̿̊, W̿̋, W̿̌, W̿̍, W̿̎, W̿̏, W̿̐, W̿̑, W̿̒, W̿̓, W̿̔, W̿̕, W̖̿, W̗̿, W̘̿, W̙̿, W̿̚, W̛̿, W̜̿, W̝̿, W̞̿, W̟̿, W̠̿, W̡̿, W̢̿, Ẉ̿, W̤̿, W̥̿, W̦̿, W̧̿, W̨̿, W̩̿, W̪̿, W̫̿, W̬̿, W̭̿, W̮̿, W̯̿, W̰̿, W̱̿, W̲̿, W̳̿, W̴̿, W̵̿, W̶̿, W̷̿, W̸̿, W̹̿, W̺̿, W̻̿, W̼̿, W̿̽, W̿̾, W̿̿, W̿̿̄, W̿̿̇, W̿̿̈, W̿̿̉, W̿̿̊, W̿̿̋, W̿̿̌, W̿̿̍, W̿̿̎, W̿̿̏, W̿̿̐, W̿̿̑, W̿̿̒, W̿̿̓, W̿̿̔, W̿̿̕, W̖̿̿, W̗̿̿, W̘̿̿, W̙̿̿, W̿̿̚, W̛̿̿, W̜̿̿, W̝̿̿, W̞̿̿, W̟̿̿, W̠̿̿, W̡̿̿, W̢̿̿, Ẉ̿̿, W̤̿̿, W̥̿̿, W̦̿̿, W̧̿̿, W̨̿̿, W̩̿̿, W̪̿̿, W̫̿̿, W̬̿̿, W̭̿̿, W̮̿̿, W̯̿̿, W̰̿̿, W̱̿̿, W̲̿̿, W̳̿̿, W̴̿̿, W̵̿̿, W̶̿̿, W̷̿̿, W̸̿̿, W̹̿̿, W̺̿̿, W̻̿̿, W̼̿̿, W̿̿̽, W̿̿̾, W̿̿̿, W̿̿̿̄, W̿̿̿̇, W̿̿̿̈, W̿̿̿̉, W̿̿̿̊, W̿̿̿̋, W̿̿̿̌, W̿̿̿̍, W̿̿̿̎, W̿̿̿̏, W̿̿̿̐, W̿̿̿̑, W̿̿̿̒, W̿̿̿̓, W̿̿̿̔, W̿̿̿̕, W̖̿̿̿, W̗̿̿̿, W̘̿̿̿, W̙̿̿̿, W̿̿̿̚, W̛̿̿̿, W̜̿̿̿, W̝̿̿̿, W̞̿̿̿, W̟̿̿̿, W̠̿̿̿, W̡̿̿̿, W̢̿̿̿, Ẉ̿̿̿, W̤̿̿̿, W̥̿̿̿, W̦̿̿̿, W̧̿̿̿, W̨̿̿̿, W̩̿̿̿, W̪̿̿̿, W̫̿̿̿, W̬̿̿̿, W̭̿̿̿, W̮̿̿̿, W̯̿̿̿, W̰̿̿̿, W̱̿̿̿, W̲̿̿̿, W̳̿̿̿, W̴̿̿̿, W̵̿̿̿, W̶̿̿̿, W̷̿̿̿, W̸̿̿̿, W̹̿̿̿, W̺̿̿̿, W̻̿̿̿, W̼̿̿̿, W̿̿̿̽, W̿̿̿̾, W̿̿̿̿}, % Cyr  
7951 W}, % Cyr  
7952 X = {X, X̄,  
7953 X, 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, X̿̄, X̿̇, X̿̈, X̿̉, X̿̊, X̿̋, X̿̌, X̿̍, X̿̎, X̿̏, X̿̐, X̿̑, X̿̒, X̿̓, X̿̔, X̿̕, X̖̿, X̗̿, X̘̿, X̙̿, X̿̚, X̛̿, X̜̿, X̝̿, X̞̿, X̟̿, X̠̿, X̡̿, X̢̿, X̣̿, X̤̿, X̥̿, X̦̿, X̧̿, X̨̿, X̩̿, X̪̿, X̫̿, X̬̿, X̭̿, X̮̿, X̯̿, X̰̿, X̱̿, X̲̿, X̳̿, X̴̿, X̵̿, X̶̿, X̷̿, X̸̿, X̹̿, X̺̿, X̻̿, X̼̿, X̿̽, X̿̾, X̿̿, X̿̿̄, X̿̿̇, X̿̿̈, X̿̿̉, X̿̿̊, X̿̿̋, X̿̿̌, X̿̿̍, X̿̿̎, X̿̿̏, X̿̿̐, X̿̿̑, X̿̿̒, X̿̿̓, X̿̿̔, X̿̿̕, X̖̿̿, X̗̿̿, X̘̿̿, X̙̿̿, X̿̿̚, X̛̿̿, X̜̿̿, X̝̿̿, X̞̿̿, X̟̿̿, X̠̿̿, X̡̿̿, X̢̿̿, X̣̿̿, X̤̿̿, X̥̿̿, X̦̿̿, X̧̿̿, X̨̿̿, X̩̿̿, X̪̿̿, X̫̿̿, X̬̿̿, X̭̿̿, X̮̿̿, X̯̿̿, X̰̿̿, X̱̿̿, X̲̿̿, X̳̿̿, X̴̿̿, X̵̿̿, X̶̿̿, X̷̿̿, X̸̿̿, X̹̿̿, X̺̿̿, X̻̿̿, X̼̿̿, X̿̿̽, X̿̿̾, X̿̿̿, X̿̿̿̄, X̿̿̿̇, X̿̿̿̈, X̿̿̿̉, X̿̿̿̊, X̿̿̿̋, X̿̿̿̌, X̿̿̿̍, X̿̿̿̎, X̿̿̿̏, X̿̿̿̐, X̿̿̿̑, X̿̿̿̒, X̿̿̿̓, X̿̿̿̔, X̿̿̿̕, X̖̿̿̿, X̗̿̿̿, X̘̿̿̿, X̙̿̿̿, X̿̿̿̚, X̛̿̿̿, X̜̿̿̿, X̝̿̿̿, X̞̿̿̿, X̟̿̿̿, X̠̿̿̿, X̡̿̿̿, X̢̿̿̿, X̣̿̿̿, X̤̿̿̿, X̥̿̿̿, X̦̿̿̿, X̧̿̿̿, X̨̿̿̿, X̩̿̿̿, X̪̿̿̿, X̫̿̿̿, X̬̿̿̿, X̭̿̿̿, X̮̿̿̿, X̯̿̿̿, X̰̿̿̿, X̱̿̿̿, X̲̿̿̿, X̳̿̿̿, X̴̿̿̿, X̵̿̿̿, X̶̿̿̿, X̷̿̿̿, X̸̿̿̿, X̹̿̿̿, X̺̿̿̿, X̻̿̿̿, X̼̿̿̿, X̿̿̿̽, X̿̿̿̾, X̿̿̿̿}, % Cyr  
7954 Y = {Y, Ȳ, Ẏ, Ÿ, Ỷ, Y̊, Y̋, Y̌, Y̍, Y̎, Y̏, Y̐, Y̑, Y̒, Y̓, Y̔, Y̕, Y̖, Y̗, Y̘, Y̙, Y̚, Y̛, Y̜, Y̝, Y̞, Y̟, Y̠, Y̡, Y̢, Ỵ, Y̤, Y̥, Y̦, Y̧, Y̨, Y̩, Y̪, Y̫, Y̬, Y̭, Y̮, Y̯, Y̰, Y̱, Y̲, Y̳, Y̴, Y̵, Y̶, Y̷, Y̸, Y̹, Y̺, Y̻, Y̼, Y̽, Y̾, Y̿, Y̿̄, Y̿̇, Y̿̈, Y̿̉, Y̿̊, Y̿̋, Y̿̌, Y̿̍, Y̿̎, Y̿̏, Y̿̐, Y̿̑, Y̿̒, Y̿̓, Y̿̔, Y̿̕, Y̖̿, Y̗̿, Y̘̿, Y̙̿, Y̿̚, Y̛̿, Y̜̿, Y̝̿, Y̞̿, Y̟̿, Y̠̿, Y̡̿, Y̢̿, Ỵ̿, Y̤̿, Y̥̿, Y̦̿, Y̧̿, Y̨̿, Y̩̿, Y̪̿, Y̫̿, Y̬̿, Y̭̿, Y̮̿, Y̯̿, Y̰̿, Y̱̿, Y̲̿, Y̳̿, Y̴̿, Y̵̿, Y̶̿, Y̷̿, Y̸̿, Y̹̿, Y̺̿, Y̻̿, Y̼̿, Y̿̽, Y̿̾, Y̿̿, Y̿̿̄, Y̿̿̇, Y̿̿̈, Y̿̿̉, Y̿̿̊, Y̿̿̋, Y̿̿̌, Y̿̿̍, Y̿̿̎, Y̿̿̏, Y̿̿̐, Y̿̿̑, Y̿̿̒, Y̿̿̓, Y̿̿̔, Y̿̿̕, Y̖̿̿, Y̗̿̿, Y̘̿̿, Y̙̿̿, Y̿̿̚, Y̛̿̿, Y̜̿̿, Y̝̿̿, Y̞̿̿, Y̟̿̿, Y̠̿̿, Y̡̿̿, Y̢̿̿, Ỵ̿̿, Y̤̿̿, Y̥̿̿, Y̦̿̿, Y̧̿̿, Y̨̿̿, Y̩̿̿, Y̪̿̿, Y̫̿̿, Y̬̿̿, Y̭̿̿, Y̮̿̿, Y̯̿̿, Y̰̿̿, Y̱̿̿, Y̲̿̿, Y̳̿̿, Y̴̿̿, Y̵̿̿, Y̶̿̿, Y̷̿̿, Y̸̿̿, Y̹̿̿, Y̺̿̿, Y̻̿̿, Y̼̿̿, Y̿̿̽, Y̿̿̾, Y̿̿̿, Y̿̿̿̄, Y̿̿̿̇, Y̿̿̿̈, Y̿̿̿̉, Y̿̿̿̊, Y̿̿̿̋, Y̿̿̿̌, Y̿̿̿̍, Y̿̿̿̎, Y̿̿̿̏, Y̿̿̿̐, Y̿̿̿̑, Y̿̿̿̒, Y̿̿̿̓, Y̿̿̿̔, Y̿̿̿̕, Y̖̿̿̿, Y̗̿̿̿, Y̘̿̿̿, Y̙̿̿̿, Y̿̿̿̚, Y̛̿̿̿, Y̜̿̿̿, Y̝̿̿̿, Y̞̿̿̿, Y̟̿̿̿, Y̠̿̿̿, Y̡̿̿̿, Y̢̿̿̿, Ỵ̿̿̿, Y̤̿̿̿, Y̥̿̿̿, Y̦̿̿̿, Y̧̿̿̿, Y̨̿̿̿, Y̩̿̿̿, Y̪̿̿̿, Y̫̿̿̿, Y̬̿̿̿, Y̭̿̿̿, Y̮̿̿̿, Y̯̿̿̿, Y̰̿̿̿, Y̱̿̿̿, Y̲̿̿̿, Y̳̿̿̿, Y̴̿̿̿, Y̵̿̿̿, Y̶̿̿̿, Y̷̿̿̿, Y̸̿̿̿, Y̹̿̿̿, Y̺̿̿̿, Y̻̿̿̿, Y̼̿̿̿, Y̿̿̿̽, Y̿̿̿̾, Y̿̿̿̿}, % Cyr  
7955 Y, Ȳ}, % Cyr  
7956 Z = {Z, Z̄, 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, Z̿̄, Z̿̇, Z̿̈, Z̿̉, Z̿̊, Z̿̋, Z̿̌, Z̿̍, Z̿̎, Z̿̏, Z̿̐, Z̿̑, Z̿̒, Z̿̓, Z̿̔, Z̿̕, Z̖̿, Z̗̿, Z̘̿, Z̙̿, Z̿̚, Z̛̿, Z̜̿, Z̝̿, Z̞̿, Z̟̿, Z̠̿, Z̡̿, Z̢̿, Ẓ̿, Z̤̿, Z̥̿, Z̦̿, Z̧̿, Z̨̿, Z̩̿, Z̪̿, Z̫̿, Z̬̿, Z̭̿, Z̮̿, Z̯̿, Z̰̿, Ẕ̿, Z̲̿, Z̳̿, Z̴̿, Z̵̿, Z̶̿, Z̷̿, Z̸̿, Z̹̿, Z̺̿, Z̻̿, Z̼̿, Z̿̽, Z̿̾, Z̿̿, Z̿̿̄, Z̿̿̇, Z̿̿̈, Z̿̿̉, Z̿̿̊, Z̿̿̋, Z̿̿̌, Z̿̿̍, Z̿̿̎, Z̿̿̏, Z̿̿̐, Z̿̿̑, Z̿̿̒, Z̿̿̓, Z̿̿̔, Z̿̿̕, Z̖̿̿, Z̗̿̿, Z̘̿̿, Z̙̿̿, Z̿̿̚, Z̛̿̿, Z̜̿̿, Z̝̿̿, Z̞̿̿, Z̟̿̿, Z̠̿̿, Z̡̿̿, Z̢̿̿, Ẓ̿̿, Z̤̿̿, Z̥̿̿, Z̦̿̿, Z̧̿̿, Z̨̿̿, Z̩̿̿, Z̪̿̿, Z̫̿̿, Z̬̿̿, Z̭̿̿, Z̮̿̿, Z̯̿̿, Z̰̿̿, Ẕ̿̿, Z̲̿̿, Z̳̿̿, Z̴̿̿, Z̵̿̿, Z̶̿̿, Z̷̿̿, Z̸̿̿, Z̹̿̿, Z̺̿̿, Z̻̿̿, Z̼̿̿, Z̿̿̽, Z̿̿̾, Z̿̿̿, Z̿̿̿̄, Z̿̿̿̇, Z̿̿̿̈, Z̿̿̿̉, Z̿̿̿̊, Z̿̿̿̋, Z̿̿̿̌, Z̿̿̿̍, Z̿̿̿̎, Z̿̿̿̏, Z̿̿̿̐, Z̿̿̿̑, Z̿̿̿̒, Z̿̿̿̓, Z̿̿̿̔, Z̿̿̿̕, Z̖̿̿̿, Z̗̿̿̿, Z̘̿̿̿, Z̙̿̿̿, Z̿̿̿̚, Z̛̿̿̿, Z̜̿̿̿, Z̝̿̿̿, Z̞̿̿̿, Z̟̿̿̿, Z̠̿̿̿, Z̡̿̿̿, Z̢̿̿̿, Ẓ̿̿̿, Z̤̿̿̿, Z̥̿̿̿, Z̦̿̿̿, Z̧̿̿̿, Z̨̿̿̿, Z̩̿̿̿, Z̪̿̿̿, Z̫̿̿̿, Z̬̿̿̿, Z̭̿̿̿, Z̮̿̿̿, Z̯̿̿̿, Z̰̿̿̿, Ẕ̿̿̿, Z̲̿̿̿, Z̳̿̿̿, Z̴̿̿̿, Z̵̿̿̿, Z̶̿̿̿, Z̷̿̿̿, Z̸̿̿̿, Z̹̿̿̿, Z̺̿̿̿, Z̻̿̿̿, Z̼̿̿̿, Z̿̿̿̽, Z̿̿̿̾, Z̿̿̿̿}, % a'  
7957 a = {a, ā, ǎ, ǎ̄, ǎ̇, ǎ̈, ǎ̉, ǎ̊, ǎ̋, ǎ̌, ǎ̍, ǎ̎, ǎ̏, ǎ̐, ǎ̑, ǎ̒, ǎ̓, ǎ̔, ǎ̕, ǎ̖, ǎ̗, ǎ̘, ǎ̙, ǎ̚, ǎ̛, ǎ̜, ǎ̝, ǎ̞, ǎ̟, ǎ̠, ǎ̡, ǎ̢, ạ̌, ǎ̤, ḁ̌, ǎ̦, ǎ̧, ą̌, ǎ̩, ǎ̪, ǎ̫, ǎ̬, ǎ̭, ǎ̮, ǎ̯, ǎ̰, ǎ̱, ǎ̲, ǎ̳, ǎ̴, ǎ̵, ǎ̶, ǎ̷, ǎ̸, ǎ̹, ǎ̺, ǎ̻, ǎ̼, ǎ̽, ǎ̾, ǎ̿, ǎ̿̄, ǎ̿̇, ǎ̿̈, ǎ̿̉, ǎ̿̊, ǎ̿̋, ǎ̿̌, ǎ̿̍, ǎ̿̎, ǎ̿̏, ǎ̿̐, ǎ̿̑, ǎ̿̒, ǎ̿̓, ǎ̿̔, ǎ̿̕, ǎ̖̿, ǎ̗̿, ǎ̘̿, ǎ̙̿, ǎ̿̚, ǎ̛̿, ǎ̜̿, ǎ̝̿, ǎ̞̿, ǎ̟̿, ǎ̠̿, ǎ̡̿, ǎ̢̿, ạ̌̿, ǎ̤̿, ḁ̌̿, ǎ̦̿, ǎ̧̿, ą̌̿, ǎ̩̿, ǎ̪̿, ǎ̫̿, ǎ̬̿, ǎ̭̿, ǎ̮̿, ǎ̯̿, ǎ̰̿, ǎ̱̿, ǎ̲̿, ǎ̳̿, ǎ̴̿, ǎ̵̿, ǎ̶̿, ǎ̷̿, ǎ̸̿, ǎ̹̿, ǎ̺̿, ǎ̻̿, ǎ̼̿, ǎ̿̽, ǎ̿̾, ǎ̿̿, ǎ̿̿̄, ǎ̿̿̇, ǎ̿̿̈, ǎ̿̿̉, ǎ̿̿̊, ǎ̿̿̋, ǎ̿̿̌, ǎ̿̿̍, ǎ̿̿̎, ǎ̿̿̏, ǎ̿̿̐, ǎ̿̿̑, ǎ̿̿̒, ǎ̿̿̓, ǎ̿̿̔, ǎ̿̿̕, ǎ̖̿̿, ǎ̗̿̿, ǎ̘̿̿, ǎ̙̿̿, ǎ̿̿̚, ǎ̛̿̿, ǎ̜̿̿, ǎ̝̿̿, ǎ̞̿̿, ǎ̟̿̿, ǎ̠̿̿, ǎ̡̿̿, ǎ̢̿̿, ạ̌̿̿, ǎ̤̿̿, ḁ̌̿̿, ǎ̦̿̿, ǎ̧̿̿, ą̌̿̿, ǎ̩̿̿, ǎ̪̿̿, ǎ̫̿̿, ǎ̬̿̿, ǎ̭̿̿, ǎ̮̿̿, ǎ̯̿̿, ǎ̰̿̿, ǎ̱̿̿, ǎ̲̿̿, ǎ̳̿̿, ǎ̴̿̿, ǎ̵̿̿, ǎ̶̿̿, ǎ̷̿̿, ǎ̸̿̿, ǎ̹̿̿, ǎ̺̿̿, ǎ̻̿̿, ǎ̼̿̿, ǎ̿̿̽, ǎ̿̿̾, ǎ̿̿̿}, % a'  
7958 a, ā, ǎ}, % Cyr  
7959 æ = {æ,  
7960 æ}, % Cyr  
7961 b = {b, b̄, ḃ},  
7962 c = {ç, ç̄, 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, ç̿̄, ç̿̇, ç̿̈, ç̿̉, ç̿̊, ç̿̋, ç̿̌, ç̿̍, ç̿̎, ç̿̏, ç̿̐, ç̿̑, ç̿̒, ç̿̓, ç̿̔, ç̿̕, ç̖̿, ç̗̿, ç̘̿, ç̙̿, ç̿̚, ç̛̿, ç̜̿, ç̝̿, ç̞̿, ç̟̿, ç̠̿, ç̡̿, ç̢̿, ç̣̿, ç̤̿, ç̥̿, ç̦̿, ç̧̿, ç̨̿, ç̩̿, ç̪̿, ç̫̿, ç̬̿, ç̭̿, ç̮̿, ç̯̿, ç̰̿, ç̱̿, ç̲̿, ç̳̿, ç̴̿, ç̵̿, ç̶̿, ç̷̿, ç̸̿, ç̹̿, ç̺̿, ç̻̿, ç̼̿, ç̿̽, ç̿̾, ç̿̿, ç̿̿̄, ç̿̿̇, ç̿̿̈, ç̿̿̉, ç̿̿̊, ç̿̿̋, ç̿̿̌, ç̿̿̍, ç̿̿̎, ç̿̿̏, ç̿̿̐, ç̿̿̑, ç̿̿̒, ç̿̿̓, ç̿̿̔, ç̿̿̕, ç̖̿̿, ç̗̿̿, ç̘̿̿, ç̙̿̿, ç̿̿̚, ç̛̿̿, ç̜̿̿, ç̝̿̿, ç̞̿̿, ç̟̿̿, ç̠̿̿, ç̡̿̿, ç̢̿̿, ç̣̿̿, ç̤̿̿, ç̥̿̿, ç̦̿̿, ç̧̿̿, ç̨̿̿, ç̩̿̿, ç̪̿̿, ç̫̿̿, ç̬̿̿, ç̭̿̿, ç̮̿̿, ç̯̿̿, ç̰̿̿, ç̱̿̿, ç̲̿̿, ç̳̿̿, ç̴̿̿, ç̵̿̿, ç̶̿̿, ç̷̿̿, ç̸̿̿, ç̹̿̿, ç̺̿̿, ç̻̿̿, ç̼̿̿, ç̿̿̽, ç̿̿̾, ç̿̿̿, ç̿̿̿̄, ç̿̿̿̇, ç̿̿̿̈, ç̿̿̿̉, ç̿̿̿̊, ç̿̿̿̋, ç̿̿̿̌, ç̿̿̿̍, ç̿̿̿̎, ç̿̿̿̏, ç̿̿̿̐, ç̿̿̿̑, ç̿̿̿̒, ç̿̿̿̓, ç̿̿̿̔, ç̿̿̿̕, ç̖̿̿̿, ç̗̿̿̿, ç̘̿̿̿, ç̙̿̿̿, ç̿̿̿





```

8127 {,}= { ,500},
8128 := { ,500},
8129 ;= { ,500},
8130 != { ,100},
8131 ? = { ,200},
8132 @ = {50,50},
8133 ~ = {200,250},
8134 \% = {50,50},
8135 * = {300,300},
8136 + = {250,250},
8137 - = {400,500}, % /hyphen
8138 – = {400,300}, % /endash
8139 — = {300,200}, % /emdash
8140 _ = {200,200}, % /underscore
8141 / = {200,300},
8142 /backslash = {200,300},
8143 ' = {300,400}, % /quotesingle
8144 ‘ = {500,700}, ’ = {500,600},
8145 “ = {500,300}, ” = {200,600},
8146 , = {400,400}, ,, = {400,400},
8147 ‹ = {400,400}, › = {300,500},
8148 « = {300,200}, » = {100,400},
8149 ¡ = {100, }, ¡ = {100, },
8150 ( = {300, }, ) = { ,300},
8151 < = {200,100}, > = {100,200},
8152 /braceleft = {400,200}, /braceright = {200,400},
8153 /angleleft = {400, }, /angleright = { ,400},
8154 † = {100,100},
8155 ‡ = { 80, 80},
8156 • = {200,200},
8157 · = {400,450}, % / periodcentered
8158 °C = { 80, 50},
8159 ℄ = { , 50},
8160 ° = {400,400},
8161 ™ = {100,200},
8162 © = {100,100},
8163 ® = {100,100},
8164 ª = {100,200},
8165 º = {100,200},
8166 ¹ = {200,250},
8167 º = { 50,100},
8168 ³ = { 50,100},
8169 ¬ = {200, },
8170 − = {300,300},
8171 ± = {150,200},
8172 × = {150,250},
8173 ÷ = {150,250},
8174 € = {100, },
8175 /one.oldstyle = {100,100},
8176 /two.oldstyle = { 50, 50},
8177 /three.oldstyle = { 30, 80},
8178 /four.oldstyle = { 50, 50},
8179 /seven.oldstyle = { 50, 80},
8180 Γ = { ,180}, % /Gamma
8181 Δ = {100,100}, % /Delta
8182 Θ = { 50, 50}, % /Theta
8183 Λ = {100,100}, % /Lambda
8184 % Ξ = {,}, % /Xi
8185 % Π = {,}, % /Pi
8186 Σ = { 50, 50}, % /Sigma
8187 Υ = {100,100}, % /Upsilon
8188 Φ = { 50, 50}, % /Phi
8189 Ψ = { 50, 50}, % /Psi
8190 % Ω = {,}, % /Omega
8191 }

```

```
8192
8193 \SetProtrusion
8194   [ name      = LMR-it ]
8195   { encoding = {EU1,EU2,TU},
8196     family   = Latin Modern Roman,
8197     shape    = {it,s1}      }
8198   {
8199     A = {125,100},
8200     Æ = {125,-55},
8201     B = {90,-40},
8202     C = {145,-75},
8203     D = {75, -28},
8204     E = {80,-55},
8205     F = {85,-80},
8206     G = {153,-15},
8207     H = {73,-60},
8208     I = {140,-120},
8209     IJ = {140,-80},
8210     J = {135,-80},
8211     K = {70,-30},
8212     L = {87, 40},
8213     M = {67,-45},
8214     N = {75,-55},
8215     O = {150,-30},
8216     Œ = {150,-55},
8217     P = {82,-50},
8218     Q = {150,-30},
8219     R = {75, 15},
8220     S = {90,-65},
8221     $ = {100,-20},
8222     T = {220,-85},
8223     U = {230,-55},
8224     V = {260,-60},
8225     W = {185,-55},
8226     X = {70,-30},
8227     Y = {250,-60},
8228     Z = {90,-60},
8229     a = {150,-10},
8230     b = {170,  },
8231     c = {173,-10},
8232     d = {150,-55},
8233     e = {180, },
8234     f = {  ,-250},
8235     g = {150,-10},
8236     h = {100, },
8237     i = {210, },
8238     ij = {210,-40},
8239     j = {  ,-40},
8240     k = {110,-50},
8241     l = {240,-110},
8242     m = {80, },
8243     n = {115, },
8244     o = {155, },
8245     q = {170,-40},
8246     r = {155,-40},
8247     s = {130, },
8248     t = {230,-10},
8249     u = {120, },
8250     v = {140,-25},
8251     w = {98,-20},
8252     x = {65,-40},
8253     y = {130,-20},
8254     z = {110,-80},
8255     0 = {170,-85},
8256     1 = {230,110},
```

8257 2 = {130,-70},  
 8258 3 = {140,-70},  
 8259 4 = {130,80},  
 8260 5 = {160, },  
 8261 6 = {175,-30},  
 8262 7 = {250,-150},  
 8263 8 = {130,-40},  
 8264 9 = {155,-80},  
 8265 . = { ,500},  
 8266 {,}= { ,450},  
 8267 := { ,300},  
 8268 ; = { ,300},  
 8269 & = {130,30},  
 8270 \% = {180,50},  
 8271 \* = {380,20},  
 8272 + = {180,200},  
 8273 @ = {180,10},  
 8274 ~ = {200,150},  
 8275 ( = {300, }, ) = { ,70},  
 8276 / = {100,100},  
 8277 - = {500,300}, % /hyphen  
 8278 – = {500,300}, % /endash  
 8279 — = {400,170}, % /emdash  
 8280 \_ = {100,200}, % /underscore  
 8281 ' = {300,400}, % /quotesingle  
 8282 " = {500,300},  
 8283 ‘ = {800,200}, ’ = {800,-20},  
 8284 “ = {540,100}, ” = {500,100},  
 8285 , = {300,700}, ,, = {200,600},  
 8286 ‹ = {500,300}, › = {400,400},  
 8287 « = {400,100}, » = {200,300},  
 8288 ¡ = {200, }, ì = {200, },  
 8289 < = {300,100}, > = {200,100},  
 8290 /backslash = {300,300},  
 8291 /braceleft = {400,100}, /braceright = {200,200},  
 8292 † = {200, 80},  
 8293 ‡ = {120, 80},  
 8294 • = {220,100},  
 8295 · = {550,300}, % / periodcentered  
 8296 °C = {170, },  
 8297 © = {100, 50},  
 8298 ¶ = {200, },  
 8299 ° = {500,300},  
 8300 ™ = {200, 70},  
 8301 © = { 50, 70},  
 8302 ® = { 50, 70},  
 8303 ª = {140,100},  
 8304 º = {140,100},  
 8305 ¹ = {400,150},  
 8306 º = {250, 80},  
 8307 ³ = {250, 80},  
 8308 ¬ = {250, 80},  
 8309 – = {300,200},  
 8310 ± = {150,170},  
 8311 × = {200,200},  
 8312 ÷ = {200,200},  
 8313 € = {150, },  
 8314 /one.oldstyle = {100,100},  
 8315 /two.oldstyle = {100, 80},  
 8316 /three.oldstyle = { 80, 50},  
 8317 /four.oldstyle = { 80, 80},  
 8318 /five.oldstyle = { 50, },  
 8319 /six.oldstyle = { 50, },  
 8320 /seven.oldstyle = { 80, 80},  
 8321 /eight.oldstyle = { 50, },

```

8322   Γ = {100,120}, % /Gamma
8323   Δ = {120,100}, % /Delta
8324   Θ = {120, 50}, % /Theta
8325   Λ = {130,100}, % /Lambda
8326   Ξ = {100,},    % /Xi
8327   Π = {100,},    % /Pi
8328   Σ = {100, 50}, % /Sigma
8329   Υ = {180,100}, % /Upsilon
8330   Φ = {130, 70}, % /Phi
8331   Ψ = {130, 50}, % /Psi
8332   Ω = { 50,},    % /Omega
8333   }
8334   /LatinModernRoman
8335   (*CharisSIL)
8336   \SetProtrusion
8337   [ name      = Charis-default ]
8338   { encoding = {EU1,EU2,TU},
8339     family   = Charis SIL }
8340   {
8341   A = {50,50},
8342   Æ = {50,50},
8343   C = {50, },
8344   D = { ,50},
8345   F = { ,50},
8346   G = {50, },
8347   J = {100, },
8348   K = { ,50},
8349   L = { ,50},
8350   Ḷ = { ,100},
8351   O = {50,50},
8352   Œ = {50, },
8353   P = { ,50},
8354   Q = {50,70},
8355   R = { ,50},
8356   ß = { ,40}, % capital sharp s
8357   T = {50,50},
8358   V = {50,50},
8359   W = {50,50},
8360   X = {50,50},
8361   Y = {50,50},
8362   k = { ,50},
8363   ḷ = { ,150},
8364   r = { ,50},
8365   t = { ,50},
8366   v = {50,50},
8367   w = {50,50},
8368   x = {50,50},
8369   y = { ,50},
8370   1 = {150,150},
8371   2 = {50,50},
8372   3 = {50, },
8373   4 = {100,50},
8374   6 = {50, },
8375   7 = {50,80},
8376   9 = {50,50},
8377   . = { ,600},
8378   {,} = { ,500},
8379   : = { ,400},
8380   ; = { ,300},
8381   ! = { ,100},
8382   ? = { ,200},
8383   @ = {50,50},
8384   ~ = {200,250},
8385   \% = { ,50},
8386   * = {300,300},

```

8387 + = {200,250},  
8388 / = { ,200},  
8389 /backslash = {150,200},  
8390 | = {200,200},  
8391 - = {400,500}, % hyphen  
8392 – = {200,300}, % endash  
8393 — = {150,250}, % emdash  
8394 ⎯ = {200,200}, % Horizontal Bar = \texttwelveudash  
8395 - = {150,150}, % Figure Dash = \textthreequartersemdash  
8396 \_ = {100,100},  
8397 {=} = {100,100},  
8398 ‘ = {300,400}, ’ = {300,400},  
8399 “ = {300,300}, ” = {300,300},  
8400 , = {400,400}, „ = {300,300},  
8401 ‹ = {400,300}, › = {300,400},  
8402 ‹‹ = {200,200}, ›› = {150,300},  
8403 ¡ = {100, }, ¿ = {100, },  
8404 ( = {200, }, ) = { ,200},  
8405 < = {200,150}, > = {100,200},  
8406 [ = {100, }, ] = { ,100},  
8407 /braceleft = {200, }, /braceright = { ,300},  
8408 † = { 80, 80},  
8409 ‡ = {100,100},  
8410 • = {200,200},  
8411 ° = {150,200},  
8412 ™ = {150,150},  
8413 ¢ = { 50, },  
8414 £ = { 50, },  
8415 † = {200,200},  
8416 © = {100,100},  
8417 ® = {100,100},  
8418 º = {100,200},  
8419 ° = {200,200},  
8420 ¬ = {200, 50},  
8421 μ = { ,100},  
8422 ¶ = { ,100},  
8423 · = {300,400},  
8424 <sup>1</sup> = {200,300},  
8425 <sup>2</sup> = {100,200},  
8426 <sup>3</sup> = {100,200},  
8427 € = {100, },  
8428 ± = {150,200},  
8429 × = {200,200},  
8430 ÷ = {250,250},  
8431 /minus = {200,200},  
8432 − = {200,200},  
8433 % Cyrillic  
8434 Б = { ,50},  
8435 Г = { ,130},  
8436 Ж = {50,50},  
8437 З = {30,50},  
8438 Л = {50, },  
8439 У = {50,50},  
8440 Ф = {50,50},  
8441 Ч = {100, },  
8442 Ъ = { ,50},  
8443 Ь = { ,50},  
8444 Э = {50,50},  
8445 Ю = { ,40},  
8446 Я = {50, },  
8447 В = {50,50},  
8448 € = {50, },  
8449 Ъ = {50,100},  
8450 € = {50, },  
8451 Ъ = {50,50},

```

8452   Ѓ = { ,50},
8453   Є = {50,50},
8454   Ѕ = {100,100},
8455   І = {50,50},
8456   Ї = { ,50},
8457   Љ = { ,50},
8458   Ћ = {50,80},
8459   Ќ = { ,80},
8460   Ѝ = {50,50},
8461   Ў = {50, },
8462   Џ = {50,40},
8463   Й = { ,50},
8464   Р = {50, },
8465   С = { ,50},
8466   Т = { ,50},
8467   У = { ,100},
8468   Ф = {50,50},
8469   Г = { ,70},
8470   К = { ,50},
8471   Л = {50, },
8472   Т = {50,50},
8473   Ф = {50,50},
8474   Ч = {50, },
8475   Ъ = { ,50},
8476   Ь = { ,50},
8477   Э = { ,50},
8478   Я = {50, },
8479   Ь = {50, },
8480   Ъ = { ,50},
8481   Ь = { ,50},
8482   V = {50,50},
8483   Ч = {50, },
8484   Ъ = { ,50},
8485   Y = {50,50},
8486   Ъ = { ,50},
8487   Ъ = { ,50},
8488   У = { ,100},
8489   ъ = {100,100},
8490   ʒ = {50,50},
8491   ь = {50,70},
8492   э = { ,70},
8493   ю = {50,30},
8494   я = { ,50},
8495   ѐ = { ,50},
8496   % Д П Ц Ш Щ Ъ Ы Ь Џ Ѡ ѡ Ѣ ѣ Ѥ
8497   % в д ж з и м н п ц ш ы ю ѣ ѥ Ѧ ѧ Ѩ ѩ Ѫ ѫ Ѭ ѭ
8498   % Greek
8499   Δ = {50,50},
8500   Ψ = {50,50},
8501   γ = {70,70},
8502   λ = {40,70},
8503   π = {40,50},
8504   ρ = { ,50},
8505   σ = { ,50},
8506   χ = {50,50},

8507 }
8508
8509 \SetProtrusion
8510 [ name = Charis-it ]
8511 { encoding = {EU1,EU2,TU},
8512   family = Charis SIL,
8513   shape = {it,sl} }
8514 {
8515   C = {50, },
8516   G = {50, },

```

8517 J = {50, },  
8518 L = {50,50},  
8519 O = {50, },  
8520 Œ = {50, },  
8521 Q = {50, },  
8522 S = {50, },  
8523 \$ = {50, },  
8524 T = {70, },  
8525 o = {50,50},  
8526 p = { ,50},  
8527 q = {50, },  
8528 t = { ,50},  
8529 w = { ,50},  
8530 y = { ,50},  
8531 1 = {150,100},  
8532 3 = {50, },  
8533 4 = {100, },  
8534 6 = {50, },  
8535 7 = {100, },  
8536 . = { ,700},  
8537 {,} = { ,600},  
8538 : = { ,400},  
8539 ; = { ,400},  
8540 ? = { ,150},  
8541 & = { ,80},  
8542 \% = {50,50},  
8543 \* = {300,200},  
8544 + = {250,250},  
8545 @ = {80,50},  
8546 ~ = {150,150},  
8547 / = { ,150},  
8548 /backslash = {150,150},  
8549 - = {300,400}, % hyphen  
8550 - = {200,300}, % endash  
8551 — = {150,200}, % emdash  
8552 \_ = { ,100},  
8553 {=} = {200,200},  
8554 ± = {150,200},  
8555 × = {250,250},  
8556 ÷ = {250,250},  
8557 ° = {150,200},  
8558 · = {300,400},  
8559 ‘ = {400,200}, ’ = {400,200},  
8560 “ = {300,200}, ” = {400,200},  
8561 , = {200,500}, „ = {150,500},  
8562 ‹ = {300,400}, › = {200,500},  
8563 « = {200,300}, » = {150,400},  
8564 ( = {200, }, ) = { ,200},  
8565 < = {200,200}, > = {200,200},  
8566 /braceleft = {300, }, /braceright = { ,200},  
8567 % Cyrillic  
8568 Ж = {50,30},  
8569 Л = {50, },  
8570 У = {50,30},  
8571 Ф = {50, },  
8572 Ч = {100, },  
8573 Ъ = { ,50},  
8574 Ь = { ,50},  
8575 Э = {50,50},  
8576 Я = {50, },  
8577 В = {50,50},  
8578 Ъ = {50,50},  
8579 Ъ = {140,100},  
8580 Ъ = {70,50},  
8581 Ъ = {50,80},

```

8582   Ḥ = { ,80},
8583   Ŧ = {50,50},
8584   Γ = {50,50},
8585   Δ = {50,30},
8586   Μ = {50, },
8587   Φ = {50, },
8588   Ч = {50, },
8589   Ъ = { ,50},
8590   Ь = { ,50},
8591   Э = { ,50},
8592   Я = {50, },
8593   Љ = {50,50},
8594   Њ = { ,50},
8595   V = {50,50},
8596   Ъ = { ,50},
8597   Յ = {140,100},
8598   Ճ = {70,50},
8599   Ն = {50,70},
8600   Ӧ = { ,70},
8601   % Greek
8602   Γ = { ,130},
8603   Δ = {50,50},
8604   Ψ = {50,50},
8605   γ = {70,70},
8606   λ = {40,70},
8607   π = {40,50},
8608   ρ = { ,50},
8609   σ = { ,50},
8610   χ = {50,50},
8611   }

```

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).

```

8612
8613 % quick and dirty -- maybe we'll promote this to a
8614 % regular key some time
8615 \define@key{MT@pr@c}{command}{\csname #1\endcsname}
8616
8617 % glyph names have changed with version 5.0 of Charis SIL:
8618 % before: /a.SC, /b.SC, ...
8619 % after: /a.sc, /b.sc, ...
8620 \ifx\MT@lua\undefined
8621   \gdef\MT@get@CHARIS@SC{
8622     % test whether glyph "a.sc" exists
8623     \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
8624       \gdef\MT@CHARIS@SC{sc}%
8625     \else
8626       \gdef\MT@CHARIS@SC{SC}%
8627     \fi
8628   }
8629 \else
8630   \gdef\MT@get@CHARIS@SC{
8631     \gdef\MT@CHARIS@SC{\MT@lua{
8632       % check font version
8633 % -- why doesn't this work?:
8634 %   f = font.getfont(font.current());
8635 %   i = fontloader.info(f.filename);
8636 %   if (tonumber(i.version) < 5) then;
8637 %   if (tonumber(fontloader.info(font.getfont(font.current()).filename).version) < 5) then;
8638 %     tex.print("SC");
8639 %   else;
8640 %     tex.print("sc");
8641 %   end

```

```

8642   }}
8643   }
8644 \fi
8645
8646 \SetProtrusion
8647   [ name      = Charis-sc,
8648     load      = Charis-default,
8649     command   = {MT@get@CHARIS@SC} ]
8650   { encoding = {EU1,EU2,TU},
8651     family   = Charis SIL,
8652     shape    = {sc} }
8653   {
8654     % A = {100,100}, % etc., doesn't work with \textsc
8655     /a.\MT@CHARIS@SC = {100,100},
8656     /c.\MT@CHARIS@SC = {50, },
8657     /d.\MT@CHARIS@SC = { ,50},
8658     /f.\MT@CHARIS@SC = { ,50},
8659     /g.\MT@CHARIS@SC = {50, },
8660     /j.\MT@CHARIS@SC = {100, },
8661     /k.\MT@CHARIS@SC = { ,50},
8662     /l.\MT@CHARIS@SC = { ,50},
8663     /f.l.\MT@CHARIS@SC = { ,50},
8664     /o.\MT@CHARIS@SC = {50,50},
8665     /oe.\MT@CHARIS@SC = {50, },
8666     /q.\MT@CHARIS@SC = {50,70},
8667     /r.\MT@CHARIS@SC = { ,50},
8668     /t.\MT@CHARIS@SC = {50,100},
8669     /v.\MT@CHARIS@SC = {50,50},
8670     /w.\MT@CHARIS@SC = {50,50},
8671     /x.\MT@CHARIS@SC = {50,50},
8672     /y.\MT@CHARIS@SC = {50,50}
8673   }
8674 </CharisSIL>
8675 < *PalatinoLinotype >
8676 \SetProtrusion
8677   [ name      = palatino-default ]
8678   { encoding = {EU1,EU2,TU},
8679     family   = {PalatinoLinotype} }
8680   {
8681     A = {50,50},
8682     D = { ,50},
8683     J = {50, },
8684     K = { ,50},
8685     L = { ,50},
8686     O = {25, },
8687     T = {50,50},
8688     V = {50,50},
8689     W = {50,50},
8690     X = {50,50},
8691     Y = {50,50},
8692     b = { ,25},
8693     d = {25,30},
8694     f = { ,50},
8695     g = { ,100},
8696     k = { ,50},
8697     p = { ,50},
8698     q = {50, },
8699     r = { ,50},
8700     t = { ,50}, ◆ = { ,50}, ◆ = { ,50},
8701     v = {75,50},
8702     w = {50,50},
8703     x = {50,50},
8704     y = {50,70},
8705     1 = {100,50},

```

```

8706 2 = {25,50},
8707 4 = {50, },
8708 6 = {50, },
8709 9 = {25, },
8710 Æ = {100, },
8711 Œ = {25, },
8712 . = { ,700}, .. = { ,350}, ... = { ,150},
8713 {,} = { ,500},
8714 := { ,500},
8715 ; = { ,500},
8716 ! = { ,100}, !! = { ,100},
8717 ? = { ,200}, ?? = { ,200},
8718 @ = {50,50},
8719 ~ = {200,250},
8720 & = {50,100},
8721 \% = {100,100},
8722 * = {200,200},
8723 + = {250,250},
8724 ( = {100, }, ) = { ,300},
8725 / = {200,300},
8726 - = {400,500},
8727 \textendash = {300,300}, \textemdash = {200,200},
8728 \textquoteleft = {500,700}, \textquoteright = {500,700},
8729 \textquotedblleft = {300,400}, \textquotedblright = {300,400},
8730 \textbackslash = {200,300},
8731 \quotesinglbase = {400,400}, \quotedblbase = {400,400},
8732 \guilsinglleft = {400,400}, \guilsinglright = {300,500},
8733 \guillemotleft = {300,300}, \guillemotright = {200,400},
8734 \textexclamdown = {100, }, \textquestiondown = {100, },
8735 \textbraceleft = {400,200}, \textbraceright = {200,400},
8736 \textless = {200,100}, \textgreater = {100,200},
8737 ≤ = {200,100}, ≥ = {100,200},
8738 \textminus = {300,300},
8739 \texttrademark = {200,200},
8740 \textcopyright = {200,200},
8741 \textregistered = {200,200},
8742 \textdegree = {300,300},
8743 ¡ = {450,500}, ¬ = {250,150},
8744 ◆ = {150,250},
8745 · = {850, 700},
8746 ¶ = {100,0},
8747 × = {150, 300},
8748 ª = {300,300}, º = {300,300},
8749 ° = {200,400},
8750 ¹ = {400,350}, º = {200,300},  = {250,400},
8751 ¼ = {250,350}, ½ = {200,300}, ¾ = {250,400},
8752 ⅞ = {200,450}, ⅝ = {250,400}, ⅜ = {200,350},
8753 ⅓ = {200,400},
8754 ⅔ = {400,250}, ⅖ = {200,300}, ⅜ = {250,400},
8755 ⅘ = {250,350}, ⅙ = {200,300}, ⅚ = {250,400},
8756 ⅞ = {200,450}, ⅝ = {250,400}, ⅜ = {200,350},
8757 ± = {150,100}, ÷ = {300,300},
8758 þ = { ,25},
8759 ₠ = {300,450}, ₡ = {300,450},
8760 ₢ = {300,450}, ₣ = {300,450},
8761 ₤ = {200,250}, ₦ = {200,250},
8762 π = {50, },
8763 f = { ,50},
8764 № = {100,150},
8765 \textservicemark = {100,200},
8766 - = {400,500}, - = {400,500}, - = {200,300},
8767 - = {205,305}, - = {200,300}, - = {50,150},
8768 • = {125,200},
8769 % /a.sc = {50,50},
8770 }

```

```

8771
8772 \SetProtrusion
8773   [ name      = palatino-it  ]
8774   { encoding = {EU1,EU2,TU},
8775     family   = {PalatinoLinotype},
8776     shape    = {it,s1} }
8777   {
8778     A = {50,50},
8779     Æ = {50, },
8780     B = {50, },
8781     C = {50, },
8782     D = {50,50},
8783     E = {50, },
8784     F = {50, },
8785     G = {50, },
8786     H = {50, },
8787     K = {50, },
8788     L = {50, },
8789     O = {50, },
8790     Œ = {50, },
8791     P = {50, },
8792     Q = {50, },
8793     R = {50, },
8794     S = {50, },
8795     $ = {50, },
8796     T = {100, },
8797     U = {50, },
8798     V = {100,50},
8799     W = {50, },
8800     X = {50, },
8801     Y = {100,50},
8802     b = { ,50},
8803     c = {25, },
8804     g = {75, },
8805     i = {25, },
8806     m = { ,50},
8807     n = { ,50},
8808     p = { ,25},
8809     q = {25, },
8810     x = { ,50},
8811     1 = {100, },
8812     2 = {50, },
8813     4 = {50, },
8814     7 = {50, },
8815     . = { ,500},    .. = { ,350},    ... = { ,200},
8816     {,} = { ,500},
8817     := { ,300},
8818     ; = { ,300},
8819     ? = { ,300},    ʔ = { ,300},
8820     & = {50,50},
8821     \% = {100,100},
8822     * = {200,200},
8823     + = {150,200},
8824     @ = {50,50},
8825     ~ = {200,150},
8826     ( = {200, },    ) = { ,200},
8827     / = {100,200},
8828     - = {300,500},
8829     \textendash = {300,300}, \textemdash = {200,200},
8830     \textquoteleft = {700,400}, \textquoteright = {700,400},
8831     \textquotedblleft = {500,300}, \textquotedblright = {500,300},
8832     _ = {100,100},
8833     \textbackslash = {100,200},
8834     \quotesinglbase = {500,500}, \quotedblbase = {400,400},
8835     \guilsinglleft = {400,400}, \guilsinglright = {300,500},

```

```

8836 \guillemotleft = {300,300}, \guillemotright = {300,300},
8837 \textexclamdown = {100, }, \textquestiondown = {200, },
8838 \textbraceleft = {200,100}, \textbraceright = {200,200},
8839 \textless = {300,100}, \textgreater = {200,100},
8840 ≤ = {200,100}, ≥ = {100,200},
8841 † = {450,500}, ‡ = {250,150},
8842 · = {850,700},
8843 ¶ = {100,0},
8844 × = {150,300},
8845 ª = {300,250}, º = {300,300}, ° = {300,250},
8846 º = {300,200},
8847 ¹ = {300,150}, ² = {350,200}, ³ = {250,150},
8848 ⁴ = {350,100}, ⁵ = {300,50}, ⁶ = {400,100},
8849 ⁷ = {400,50}, ⁸ = {250,50}, ⁹ = {300,50},
8850 ₀ = {300,300},
8851 ₁ = {300,350}, ₂ = {300,150}, ₃ = {250,250},
8852 ₄ = {400,200}, ₅ = {300,100}, ₆ = {450,200},
8853 ₇ = {450,150}, ₈ = {400,250}, ₉ = {400,200},
8854 ± = {150,100}, ÷ = {300,300},
8855 þ = {50, },
8856 † = {250,200}, ‡ = {250,200},
8857 ˙ = {300,450}, ˘ = {300,450},
8858 ˙ = {300,450}, ˘ = {300,450},
8859 - = {300,500}, - = {300,500}, - = {100,300},
8860 - = {125,305}, - = {200,300}, - = {125,150},
8861 • = {125,200}
8862 }
8863
8864 \SetProtrusion
8865 [ name = palatino-sc,
8866 load = palatino-default ]
8867 { encoding = {EU1,EU2,TU},
8868 family = {PalatinoLinotype},
8869 shape = sc }
8870 {
8871 a = {50,50},
8872 æ = {50, },
8873 b = {0,0},
8874 d = {0,0},
8875 f = {0,0},
8876 g = {0,0},
8877 j = {50, },
8878 l = { ,50},
8879 o = {0,0},
8880 p = {0,0},
8881 q = {0, },
8882 r = { ,0},
8883 t = {50,50},
8884 y = {50,50},
8885 fl = {0,50},
8886 ffl = {0,50},
8887 ◊ = {0,50},
8888 ◊ = {0,50}
8889 }
8890 /PalatinoLinotype
8891

```

## 17 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.

```

8892 (*test)
8893 \documentclass{article}
8894
8895 %% Here you can specify the font you want to test, using
8896 %% the commands \fontfamily, \fontseries and \fontshape.
8897 %% Make sure to end all lines with a comment character!
8898 \newcommand*\TestFont{%
8899   \fontfamily{ppl}%
8900   %% \fontseries{b}%
8901   %% \fontshape{it}% sc, sl
8902 }
8903
8904 \usepackage{ifthen}
8905 \usepackage[T1]{fontenc}
8906 \usepackage[latin1]{inputenc}
8907 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
8908
8909 \pagestyle{empty}
8910 \setlength{\parindent}{0pt}
8911 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
8912 \newcommand*\testprotrusion[2][ ]{%
8913   \ifthenelse{\equal{#1}{r}}{\#2}%
8914   lorem ipsum dolor sit amet,
8915   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
8916   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
8917   you know the rest%
8918   \ifthenelse{\equal{#1}{l}}{\#2}%
8919   \linebreak
8920   {\fontencoding{\encodingdefault}%
8921   \fontseries{\seriesdefault}%
8922   \fontshape{\shapedefault}%
8923   \selectfont
8924   Here is the beginning of a line, \dotfill and here is its end}\linebreak
8925 }
8926 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
8927 \def\stripprefix#1>{}
8928 \newcount\charcount
8929 \begin{document}
8930
8931 \microtypesetup{expansion=false}
8932
8933 {\centering The font in this document is called by:\\
8934 \texttt{\showTestFont}\par}\bigskip
8935
8936 \TestFont\selectfont
8937 This line intentionally left empty\linebreak
8938 %% A -- Z
8939 \charcount=65
8940 \loop
8941   \testprotrusion{\char\charcount}
8942   \advance\charcount 1
8943   \ifnum\charcount < 91 \repeat
8944 %% a -- z
8945 \charcount=97
8946 \loop
8947   \testprotrusion{\char\charcount}
8948   \advance\charcount 1
8949   \ifnum\charcount < 123 \repeat
8950 %% 0 -- 9
8951 \charcount=48
8952 \loop

```

```
8953 \testprotrusion{\char\charcount}
8954 \advance\charcount 1
8955 \ifnum\charcount < 58 \repeat
8956 %%
8957 \testprotrusion[r]{,}
8958 \testprotrusion[r]{.}
8959 \testprotrusion[r]{;}
8960 \testprotrusion[r]{:}
8961 \testprotrusion[r]{?}
8962 \testprotrusion[r]{!}
8963 \testprotrusion[l]{\textexclamdown}
8964 \testprotrusion[l]{\textquestiondown}
8965 \testprotrusion[r]{\}
8966 \testprotrusion[l]{\{
8967 \testprotrusion{/}
8968 \testprotrusion{\char~\}
8969 \testprotrusion{-}
8970 \testprotrusion{\textendash}
8971 \testprotrusion{\textemdash}
8972 \testprotrusion{\textquoteleft}
8973 \testprotrusion{\textquoteright}
8974 \testprotrusion{\textquotedblleft}
8975 \testprotrusion{\textquotedblright}
8976 \testprotrusion{\quotesinglbase}
8977 \testprotrusion{\quotedblbase}
8978 \testprotrusion{\guilsinglleft}
8979 \testprotrusion{\guilsinglright}
8980 \testprotrusion{\guillemotleft}
8981 \testprotrusion{\guillemotright}
8982
8983 \newpage
8984 The following displays the current font stretched by 5%,
8985 normal, and shrunk by 5%:
8986
8987 \bigskip
8988 \newlength{\MTln}
8989 \newcommand*\teststring
8990 {ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
8991 \settowidth{\MTln}{\teststring}
8992 \microtypesetup{expansion=true}
8993
8994 \parbox{1.05\MTln}{\teststring\linebreak\}
8995 \parbox{0.95\MTln}{\teststring}\par\bigskip
8996 \parbox{0.95\MTln}{\teststring}
8997
8998 \end{document}
8999 /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}
```

9000 *(\*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`.

9001 `\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.

9002 `\input bbox.sty`

`\tempdim` Allocate some `dimen` registers.

9003 `\newdimen\tempdim`

`\fboxrulei` Frame width of the box as  $\TeX$  sees it.

9004 `\newdimen\fboxrulei`

9005 `\fboxrulei=0.1pt`

`\fboxruleii` Frame width of the bounding box.

9006 `\newdimen\fboxruleii`

9007 `\fboxruleii=0.1pt`

`\kernboxheight` Height of the box indicating the kern.

9008 `\newdimen\kernboxheight`

9009 `\kernboxheight=5pt`

`\scalettoem` An auxiliary macro. Return a dimension relative to the `em`-width of the font. Requires `e-TeX`.

9010 `\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).

9011 `\fontinstcc`

9012 `\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.

9013 `\ifdim\fontdimen6\font = 0pt`

9014 `\typeout{***-Warning:-no-fontdimen-6-specified-***^J%}`

9015 `***-setting-it-to-\pdffontsize\font \ifnum\pdfversion < 130 pt\fi-***}`

9016 `\fontdimen6\font=\pdffontsize\font \ifnum\pdfversion < 130 pt\fi\relax`

9017 `\fi`

9018 `\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

```

9019   \input_metrics{}{\logofont,\metrics\printbbs{#1}\relax}
9020   \endinstallfonts
9021 }
9022 \normalcc
      Layers.
9023 \makeatletter
9024 \def\mtl@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
9025 \ifx\mt@objects\undefined\let\mt@objects\@empty\fi
9026 \ifx\mt@order\undefined\let\mt@order\@empty\fi
9027 \xdef\mt@order{\mt@order[(Logo)]}
9028 \let\mtl@resources\@empty
9029 \def\mtl@register#1{%
9030   \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
9031   \expandafter\xdef\csname mtl@#1\endcsname{\the\pdflastobj\space 0 R }
9032   \xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
9033   \xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
9034   \xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
9035 \mtl@register{canvas}
9036 \mtl@register{characters}
9037 \mtl@register{bounding-boxes}
9038 \mtl@register{TeX-boxes}
9039 \xdef\mt@order{\mt@order]}
9040 \global\let\mtl@objects\mt@objects
9041 \ifx\pdfcolorstack\undefined
9042   \pdfcatalog{/OCProperties <<
9043     /OCGs [\mt@objects]
9044     /D << /Order [\mt@order] >> >>}
9045 \fi
9046 \def\togglelayer#1#2{%
9047   \pdfstartlink width \wd\logobox height \ht\logobox depth \dp\logobox
9048   user{/Subtype/Link
9049     /BS << /Type/Border/W 0 >> /H/0
9050     /A << /S/SetOCGState
9051     /State[/Toggle \csname mtl@#1\endcsname] >>
9052   }#2\pdfendlink
9053 }

```

\printbbs Preparation.

```

9054 \setcommand\printbbs#1{%
9055   \setbox0\hbox{#1}%
9056   \leavevmode
9057   \kern-\fboxrulei
      The canvas in the natural width of the text minus protrusion, in color bgcolor.
9058   \mtl@layer{canvas}{%
9059     \getboundarychars#1\relax
9060     \tempdim=\dimexpr\wd0 - (\scaletoe{\lpcode\font\firstchar}+
9061       \scaletoe{\rprcode\font\lastchar})\relax
9062     \kern\dimexpr\scaletoe{\lpcode\font\firstchar}\relax
9063     \lower\dimexpr\dp0+0.05em \relax \vbox{\color{bgcolor}%
9064       \hrule width \tempdim
9065         height \dimexpr\dp0+\ht0+0.15em\relax}%
9066     \kern-\tempdim
      The baseline, in color blcolor.
9067     \vbox{\color{blcolor}%
9068       \hrule width \tempdim
9069         height \fboxrulei}%
9070     }%
9071     \kern-\dimexpr\wd0 -\scaletoe{\rprcode\font\lastchar}\relax
      The string.
9072     \printbbs #1\relax\relax
9073 }

```

\getboundarychars Get first ....

```

9074 \def\getboundarychars#1#2\relax{%

```

```

9075 \def\firstchar{~#1}%
9076 \getlastchar#1#2\relax
9077 }
\getlastchar ... and last character.
9078 \def\getlastchar#1#2{%
9079 \ifx\relax#2\relax
9080 \def\lastchar{~#1}%
9081 \else
9082 \expandafter\getlastchar
9083 \fi #2%
9084 }
\printbss Loop over all characters of the string.
9085 \def\printbss#1#2#3\relax{%
9086 \ifx\relax#1\relax
9087 \else
9088 \ifx\relax#2\relax
9089 \printbb{#1}{}%
9090 \else
9091 \printbb{#1}{#2}%
9092 \fi
9093 \expandafter\printbss
9094 \fi #2#3\relax
9095 }
\printbb Record the kern between the current and the following character, then print the character. \kerning is a fontinst
command.
9096 \setcommand\printbb#1#2{%
9097 \setbox0\hbox{\kerning{#1}{#2}\xdef\thekern{\number\result}}%
9098 \showboxes{#1}%
This could be another application.
9099 % \quad
9100 % w: \the\scaletoe{\width{#1}},
9101 % bb: \the\scaletoe{\bbleft{#1}}/%
9102 % \the\scaletoe{\bbright{#1}},
9103 % \the\scaletoe{\number\numexpr\width{#1}-\bbright{#1}\relax}
9104 % h: \height{#1}/\bbtop{#1}, \bbbottom{#1}/\depth{#1}\par
9105 }
\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' ≠ ',').
9106 \setcommand\showboxes#1{%
9107 \leavevmode
9108 \color{texcolor}%
We have to record the width of the glyph.
9109 \setbox0\hbox{\color{textcolor}{#1}}%
9110 \global\tempdim=\wd0\relax
9111 \kern-\fboxrulei
1. The TeX box: Print a frame in color texcolor. This frame shows the glyph as TeX sees it.
9112 \mtl@layer{TeX-boxes}{%
9113 \hbox{%
9114 \lower\dimexpr \dp0 + \fboxrulei\relax
9115 \hbox{%
9116 \vbox{%
9117 \hrule height\fboxrulei
9118 \hbox{%
9119 \vrule width\fboxrulei height \dimexpr\ht0 + 2\fboxrulei\relax
9120 \phantom{\unhcopy0}%
9121 \vrule width\fboxrulei
9122 }%
9123 \hrule height\fboxrulei}}}%
9124 }%
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.

```

```
9125 \kern-\wd0
9126 \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.
```

```
9127 \kern\dimexpr\scaletoeom{\bbleft{#1}}-\tempdim-\fboxruleii\relax
```

3. *The bounding box*: will be printed in color `bbcolor`.

```
9128 \mtl@layer{bounding-boxes}{%
9129   {\color{bbcolor}%
9130   \hbox{%
9131     \lower\dimexpr-\scaletoeom{\bbbottom{#1}}+\fboxruleii\relax
9132     \hbox{%
9133       \vbox{%
9134         \hrule height\fboxruleii
9135         \hbox to \dimexpr\scaletoeom{\numexpr
9136           \bbright{#1}-\bbleft{#1}\relax}+2\fboxruleii\relax{%
9137           \vrule height \dimexpr\scaletoeom{\numexpr
9138             \bbtop{#1}-\bbbottom{#1}\relax}%
9139             width\fboxruleii
9140             \hfill
9141             \vrule width\fboxruleii}%
9142           \hrule height\fboxruleii}}}%
9143   }%
9144   \kern-\dimexpr\fboxruleii+\fboxrulei\relax
9145   }%
```

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.

```
9146 \kern\scaletoeom{\numexpr\width{#1}-\bbright{#1}\relax}%
9147 \mtl@layer{TeX-boxes}{%
9148   {\ifnum\thekern<0
9149     \color{kerncolor}%
9150     \kern\scaletoeom{\thekern}%
9151     \lower\kernboxheight\hbox{\vrule width -\dimexpr\scaletoeom{\thekern}\relax
9152       height \kernboxheight}%
9153     \kern\scaletoeom{\thekern}%
9154   \else
9155     \color{texcolor}%
9156     \ifnum\thekern=0 \else
9157       \lower\kernboxheight
9158       \hbox{%
9159         \vbox{%
9160           \hrule height\fboxrulei
9161           \hbox{%
9162             \vrule height \kernboxheight width\fboxrulei
9163             \kern\dimexpr\scaletoeom{\thekern}-2\fboxrulei\relax
9164             \vrule width\fboxrulei
9165           }%
9166           \hrule height\fboxruleii}}}%
9167         \fi
9168       \fi
9169     }%
9170   }%
9171   % \kern-\fboxrulei
9172   }
```

```
9173 \newbox\logobox
9174 \def\printlogo{%
9175   \setbox\logobox=\hbox{\vbox{%
9176     \MakePercentComment
```

This is the Kepler MM font used in the logo.

```
9177 \def\logofont{pkpri9e10}
9178 \transformfont{\logofont}{\reencodefont{8r}{\fromafm{pkpmmri8a10}}}
9179 \font\thellogofont=\logofont\space at 82pt
```

This would load the italic Palatino font instead.

```

9180 %\def\logofont{pplri}
9181 %\transformfont{\logofont8r}{\reencodefont{8r}{\fromafm{\logofont8a}}}
9182 %\edef\logofont{\logofont8r}
9183 %\font\thelogofont=\logofont\space at 78pt

Load the font.
9184 \thelogofont

Protrusion values (overdone for didactic reasons).
9185 \lcode\font`M=96
9186 \rcode\font`e=46

Now we can generate the logo.
9187 \pdfliteral direct{/SXS gs}%
9188 \showlogo{Microtype}%
9189 % \rlap{\normalfont\normalsize\raisebox{55pt}{\footnotemark[1]}}%
9190 % \kern5pt\{3\baselineskip}
9191 % \long\def\@makefnctext##1{%
9192 % \leftskip 0pt
9193 % \parindent 0pt
9194 % \everypar{\parindent 0pt}%
9195 % \leavevmode\hbox to 15pt{\@thefnmark\hss}##1}
9196 % \footnotetext[1]{This graphic display on a
9197 % \togglelayer{canvas}{canvas} the \togglelayer{characters}{characters},
9198 % their \togglelayer{bounding-boxes}{bounding boxes}
9199 % and \togglelayer{TeX-boxes}{\TeX\ boxes}.}
9200 }%
9201 \edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}
9202 \immediate\pdfobj{<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>}%
9203 \immediate\pdfxform
9204 attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}
9205 resources {/Properties <<\mtl@resources>>}
9206 /ExtGState << /SXS \the\pdflastobj\space 0 R >> }
9207 \logobox
9208 % \vskip-2.5\baselineskip
9209 % \leavevmode
9210 % \togglelayer{characters}{%
9211 % \pdfrefxform\pdflastxform
9212 % }%
9213 \pdfannot\logodimens{%
9214 /Subtype/Widget /FT/Btn /T(Logo)
9215 %/F 4 % why did I say this?
9216 /AP << /N \the\pdflastxform\space 0 R >>
9217 /AA << /E << /S/SetOCGState /State[/Toggle \mtl@characters] >>
9218 /X << /S/SetOCGState /State[/Toggle \mtl@characters] >>
9219 /D << /S/SetOCGState /State[/Toggle \csname mtl@bounding-boxes\endcsname] >>
9220 /U << /S/SetOCGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >>
9221 >> }%
9222 \vspace{3\baselineskip}
9223 }

Our font.
9224 \pdfmapline{+pkpmri8r10 Kep1MM-It_385_575_10_ " TeXBase1Encoding ReEncodeFont " <8r.enc <pkpmri8a10.pfb}

Define colours (thered and thegreen are copied from microtype.dtx).
9225 \def\mtdefinecolors{
9226 \definecolor{thered}{rgb}{0.65,0.04,0.07}
9227 \definecolor{thegreen}{rgb}{0.06,0.44,0.08}
9228 \colorlet{texcolor}{thegreen!50} % TeX boxes
9229 \colorlet{kerncolor}{texcolor} % negative kerns
9230 \colorlet{bbcolor}{thered!50} % bounding box
9231 \colorlet{bgcolor}{black!8} % canvas
9232 \colorlet{blcolor}{black!50} % baseline
9233 \colorlet{textcolor}{black!40} % text
9234 }

Use with microtype.dtx
9235 \ifx\documentclass\@twoclasseserror

```

```

9236 \usepackage[xcdraw]{xcolor}
9237 \mtdefinicolors
9238 \else

```

## A.2 Document

Now we can start the document.

```

9239 \documentclass[10pt,a4paper]{ltxdoc}
9240 \providecommand\MakePercentComment{\relax}
9241 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
    Re-use the preamble from microtype.dtx.
9242 \usepackage{microtype-doc}
9243 \usepackage{attachfile}
9244 \makeatletter
9245 \pdfcatalog{/OCProperties << /OCGs [\mt@objects] /D << /Order [\mt@order] >> >>}
9246 \makeatother
9247 \begin{document}
    You are currently reading this.
9248 \DocInput{microtype-logo.dtx}
    And here's the logo.
9249 \vfill
9250 \begin{center}
9251   \printlogo \null
9252 \end{center}
9253 \vfill
9254 \expandafter\enddocument
9255 \fi
    That's it.
9256 /logo

```

## B The letterspacing illustration

This is `microtype-lssample.dtx`. You may treat this file in three different ways:

- compile it by itself
- `\input` it in the body of a dtx file
- `\input` it in the preamble: it then provides the commands
  - `\lssample`: prints the letterspacing illustration
  - `\anchorarrow`: anchors an arrow for layer `<#1>`
  - `\showarrow`: toggles layer `<#1>` or `<#2>`, and prints `<#2>`

The first two cases require the style file `microtype-doc.sty`, which can be generated from `microtype.ins` with:

```
\makefile{microtype-doc.sty}{docsty}
```

```

9257 \ifx\lssample\undefined
9258 *lssample

```

Upon popular request, here's how I've created the letterspacing illustration.<sup>31</sup>

31 Note that the `lssample` module will not be created when installing `microtype`. Instead, the source file `microtype-lssample.dtx` is included as an attachment in the PDF file. If your PDF reader supports this, you can [click here](#) to extract it; alternatively, you may use the `pdftk` tool.

## B.1 Macros

Rule width and image height and depth.

```
9259 \makeatletter
9260 \newdimen\lsamount
9261 \newdimen\lsrule
9262 \lsrule=0.2pt
9263 \def\lsheight{8pt}
9264 \def\lsdepth{12pt}
```

Our font (Adobe Caslon).

```
9265 \def\lsfont{\fontfamily{paca}\selectfont}
    Loop over all letters in <#2>, letterspacing them by <#1>.
9266 \def\dols#1#2{\lsamount=#1\relax \dols#2\enddols}
9267 \def\dolss#1#2\enddols{%
9268   \ifx\empty#2\empty\divide\lsamount 2\fi
9269   \ls{#1}%
9270   \ifx\empty#2\empty\else \dolss#2\enddols \fi
9271 }
```

One tikz picture for each letter.

```
9272 \def\ls#1{%
9273   \begin{tikzpicture}[remember picture,line width=\lsrule]
9274     \tikzstyle{every node}=[inner sep=0pt]
```

The bounding box.

```
9275     \mts@layer{stuff}{%
9276       \node[draw=thegrey,
9277         fill=theshade,
9278         outer sep=\lsrule,
9279         anchor=base,
9280         font=\lsfont]{\phantom{#1}};
9281     }
```

The letter.

```
9282     \node[anchor=base,font=\lsfont](#1){#1};
```

Two auxiliary coordinates.

```
9283     \path (#1.south west) ++(+.5\lsrule,-.5\lsrule) coordinate (#1L);
9284     \path (#1.base east) ++(-.5\lsrule,-\lsdepth) coordinate (#1R);
9285     \mts@layer{stuff}{%
```

Now draw the normal character width,

```
9286     \draw[color=thered!75,
9287       fill=thered!30,
9288       outer sep=\lsrule]
9289       (#1L) rectangle (#1R);
9290     \ifdim\lsamount>0pt
9291       \path (#1.base east) ++(+.5\lsamount,-6pt) coordinate (#1_ls);
9292       \path (#1R) ++(\lsamount+\lsrule,\lsdepth) coordinate (#1E);
```

and the letter space.

```
9293     \draw[color=thered,
9294       fill=thered!50,
9295       outer sep=\lsrule]
9296       (#1R) ++(+\lsrule,+0pt) rectangle (#1E);
9297     \fi
9298   }
9299 \end{tikzpicture}%
9300 \ignorespaces
9301 }
```

Draw the interword space.

```
9302 \def\lssp#1#2#3#4{%
9303   \mts@layer{stuff}{%
9304     \begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
9305       \tikzstyle{every draw}=[anchor=bottom]
9306       \coordinate(#1space) at (#2/2,\lsdepth/2);
```

```

9307 \coordinate(#1stretch) at (#2+#3/2,+0pt);
9308 \coordinate(#1shrink) at (#2-#4/2,+0pt);
9309 \draw[color=thegreen,fill=thegreen!50,use as bounding box]
9310 (0,0) rectangle ++(#2,+\lsdepth);
9311 \draw[color=thegreen,fill=thegreen!30]
9312 (+#2,-\lsrule) rectangle ++(+#3,-4pt+\lsrule);
9313 \draw[color=thegreen,fill=thegreen!50]
9314 (+#2,-\lsrule) rectangle ++(-#4,-4pt+\lsrule);
9315 \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
9316 (+#2,-2pt-.5\lsrule) -- ++(+#3,+0pt);
9317 \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!30]
9318 (+#2,-2pt-.5\lsrule) -- ++(-#4,+0pt);
9319 \end{tikzpicture}%
9320 }\ignorespaces
9321 }

```

Layers.

```

9322 \def\mts@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
9323 \def\mtsx@layer#1#2{\pdfliteral{/OC/stuff BDC /OC/#1 BDC}#2\pdfliteral{EMC EMC}}
9324 \ifx\mt@objects\undefined\let\mt@objects\empty\fi
9325 \ifx\mt@order\undefined\let\mt@order\empty\fi
9326 \xdef\mt@order{\mt@order[(Sheep)]}
9327 \let\mts@resources\empty
9328 \def\mts@register#1{%
9329 \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
9330 \expandafter\xdef\csname mts@#1\endcsname{\the\pdflastobj\space 0 R }
9331 \xdef\mt@objects{\mt@objects\csname mts@#1\endcsname}
9332 \xdef\mt@order{\mt@order\csname mts@#1\endcsname}
9333 \xdef\mts@resources{\mts@resources/#1 \csname mts@#1\endcsname}}
9334 \mts@register{stuff}
9335 \mts@register{tracking}
9336 \mts@register{ispace}
9337 \mts@register{ospace}
9338 \mts@register{istretch}
9339 \mts@register{ishrink}
9340 \mts@register{ostretch}
9341 \mts@register{oshrink}
9342 \mts@register{okern}
9343 \mts@register{ligature}
9344 \mts@register{_compatibility}
9345 \xdef\mt@order{\mt@order]}

```

Anchor point for the arrow in the code.

```

9346 \newcommand\anchorarrow[1]{%
9347 \tikz[remember picture,overlay]\node(#1_c){};}

```

Add an arrow from code to image.

```

9348 \newcommand\add@arrow[5][left]{%
9349 \tikz[remember picture,overlay,bend angle=14,looseness=0.75,>=latex]{%
9350 \mtsx@layer{#3}{\draw[->,thick,color=the#2](#4) to[bend #1] (#5);}}%
9351 }

```

Toggle layer.

```

9352 \def\toggle@layer#1#2#3{%
9353 \pdfstartlink
9354 user{/Subtype/Link
9355 /BS << /Type/Border/W 0 >> /H/0
9356 % /BS << /Type/Border/W 1 /S/D /D[4 1] >>
9357 % /C[0.7 0.7 0.7] /H/0
9358 /Contents(Click to Toggle!)
9359 /A << /S/SetOCGState
9360 /State[/Toggle \csname mts@#1\endcsname] >> }%
9361 \rlap{#2}%
9362 {\fboxsep=0pt \fboxrule=0pt
9363 \mts@layer{stuff}{%
9364 \rlap{\fcolorbox{white}{white}{\vphantom{kg}\color{the#3}#2}}}%
9365 \mts@layer{#1}{%

```

```

9366   \fcolorbox{white}{the#3!50}{\vphantom{kg}\color{white}#2}}%
9367   }%
9368   \pdfendlink
9369 }
9370 \newcommand\showarrow[2] [] {%
9371   \ifx\relax#1\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi
9372   \toggle@layer{\@tempa}{\itshape #2}}

```

The environment for our illustration.

```

9373 \def\ls@sample#1{%
9374   \parskip 4pt \parindent 0pt
9375   \par
9376   \vskip4pt
9377   {\leftskip 15pt
9378     \mt@pseudo@margin{\color{theblue}Click on the image to show the kerns
9379       and spacings involved. Click on emphasised words in the text below
9380       to reveal the relation of image and code.\strut}
9381     \mt@layer{_compatibility}{%
9382       \mt@place{\rlap{\hskip-\marginparwidth \color{white}%
9383         \vrule width\dimexpr\hsize+\marginparwidth\relax height\mt@unvdimen}}
9384       \mt@pseudo@margin{\color{thered}%
9385         If you had a \acronym{PDF} viewer that understands
9386         \acronym{PDF}\,{\smaller1.5}, you could hide the arrows selectively.}}
9387       \vskip-\mt@unvdimen}%
9388     \vskip-4pt
9389     \setlength\fbboxsep{4pt}}%
9390   \leavevmode
9391   \pdfstartlink
9392     user{/Subtype/Link
9393       /BS << /Type/Border/W 0 >> /H/0
9394       /A << /S/SetOCGState
9395         /State[/Toggle \mts@stuff] >> }%
9396     \fcolorbox{theframe}{theshade}%
9397     {\fontsize{34}{38}\selectfont #1}%
9398   \pdfendlink
9399   \par\medskip
9400 }%
9401 \def\x{\pdfpageresources{/Properties <<\mts@resources>>}}\x
9402 }

```

Now define the illustration to be used in the document.

```

9403 \def\lssample{%
9404   \ls@sample{%
9405     \dols{0pt}{Stop}
9406     \lssp{o}{0.45em}{0.25em}{0.15em}
9407     \dols{0.16em}{\st{ealing}\hskip-\dimexpr 0.08em+\lssrule\relax
9408     \lssp{i}{13.82pt}{4.65pt}{2.08pt}
9409     \dols{0.16em}{sheep}
9410     \dols{0pt}{!}
9411   }%

```

Don't forget to add the arrows.

```

9412 \vspace{-\baselineskip}
9413 \add@arrow{red} {tracking} {\lsamount_c.east} {a_ls}
9414 \add@arrow{red} {okern} {okernend_c.east} {p_ls}
9415 \add@arrow{green} {ospace} {ospace_c.east} {ospace}
9416 \add@arrow{green} {ispace} {ispace_c.center} {ispace}
9417 \add@arrow{green!75} {istretch} {istretch_c.east} {istretch.north}
9418 \add@arrow{green!75} {ishrink} {ishrink_c.west} {ishrink.north}
9419 \add@arrow{green!75} {ostretch} {ostretch_c.east} {ostretch.north}
9420 \add@arrow{green!75} {oshrink} {oshrink_c.east} {oshrink.north}
9421 \add@arrow[right]{grey}{ligature}{nolig_c.east} {st.center}
9422 }
9423 \fi

```

This is for use with microtype.dtx

```

9424 \ifx\documentclass\@twoclasseserror

```

```
9425 \usepackage{tikz}
9426 \else
```

## B.2 Document

```
9427 \documentclass[10pt,a4paper]{ltxdoc}
9428 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}

    Re-use the preamble from microtype.dtx.
9429 \usepackage{microtype-doc}
9430 \usepackage{attachfile}
9431 \usepackage{tikz}
9432 \makeatletter
9433 \pdfcatalog{/OCProperties << /OCGs [\mt@objects]
9434                /D << /Order [\mt@order] /BaseState/OFF >> >> }
9435 \makeatother
9436 \begin{document}

    You are currently reading this.
9437 \DocInput{microtype-1ssample.dtx}

    Now show what we are able to do.
9438 \noindent
9439 Since a picture is worth a thousand words, probably even more if, in our
9440 case, it depicts a couple of letterspaced words, let's bring one to sum up
9441 these somewhat confusing options. Suppose you had the following settings
9442 (which I would in no way recommend; they are only for illustrative purposes):
9443 \begin{verbatim}
9444 \SetTracking
9445 [ no ligatures = {"\anchorarrow{nolig}"f},
9446   spacing      = {60"\anchorarrow{ispace}"0*,"%
9447                 "-1"\anchorarrow{istretch}"00*,"%\anchorarrow{ishrink}"},
9448   outer spacing = {4"\anchorarrow{ospace}"50,"%
9449                 "2"\anchorarrow{ostretch}"50,1"\anchorarrow{oshrink}"50},
9450   outer kerning = {"\anchorarrow{okernbegin}"*,"%
9451                 "\anchorarrow{okernend}"*} ]
9452 { encoding = * }
9453 { 1"\anchorarrow{lsamount}"60 }
9454 \end{verbatim}
9455 and then write:
9456 \begin{verbatim}
9457 Stop \textls{stealing sheep}!
9458 \end{verbatim}
9459 this is the (typographically dubious) outcome:
9460
9461 \lssample
9462
9463 \noindent
9464 While the word `Stop' is not letterspaced, the space between the letters in
9465 the other two words is expanded by the \showarrow[tracking]{tracking-amount}{red}
9466 of 160/1000\,em\,=\allowbreak\,0.16\,em.
9467 The \showarrow[ispace]{inner~space}{green} within the letterspaced text is
9468 increased by 60%, while its \showarrow[istretch]{stretch}{green} amount is
9469 decreased by 10% and the \showarrow[ishrink]{shrink}{green} amount is left
9470 untouched.
9471 The \showarrow[ospace]{outer~space}{green} (of 0.45\,em) immediately before the
9472 piece of text may \showarrow[ostretch]{stretch}{green} by 0.25\,em and
9473 \showarrow[oshrink]{shrink}{green} by 0.15\,em.
9474 Note that there is no outer space after the text, since the exclamation mark
9475 immediately follows; instead, the default \showarrow[okern]{outer~kern}{red}
9476 of half the letterspace amount (0.08\,em) is added.
9477 Furthermore, one \showarrow[ligature]{grey} wasn't broken up, because we
9478 neglected to specify the `|s|' in the |no ligatures| key.
9479
9480 \expandafter\enddocument
9481 \fi
9482 </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 *Harald Harders*) . . . . . 86  
 remove 8-bit characters from the configuration files (suggested by *Harald Harders*) . . . . . 143  
 Protrusion: add factors for some more characters settings for Adobe Minion (contributed by *Harald Harders*) . . . . . 151  
`\DeclareCharacterInheritance`: new command: possibility to specify character inheritance . . . . . 117  
`\MT@declare@sets`: remove spaces around set name 103  
`\MT@find@file`: fix: also check whether the file for the base font family has already been loaded . . 86  
`\MT@get@basefamily`: only remove suffix if it is 'x' or 'j' . . . . . 87  
`\MT@get@listname@`: don't check for empty attributes list . . . . . 88  
`\MT@ifempty`: fix: use category code 12 for the percent character (reported by *Tom Kink*) . . . . . 45  
`\MT@is@number`: numbers may also be specified in hexadecimal or octal (suggested by *Harald Harders*) . . . . . 93  
`\MT@pdftex@no`: fix: version check (reported by *Harald Harders*) . . . . . 40  
`\MT@permute`: don't use sets for empty encoding . . 119  
`\MT@setup@expansion`: issue an error instead of a warning, when pdfTeX version is too old for `autoexpand` . . . . . 134  
`\MT@split@codes`: fix: allow zero and negative values 63  
`\MT@use@set`: remove spaces around set name . . . 108

### 2004/10/03 **Version 1.2**

Font aliases: declare `cmor` as an alias of `cmr` . . . . . 141  
 Font sets: new: `allmath` and `basicmath` . . . . . 140  
 Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding . . . . . 175  
 add settings for Computer Modern Roman math symbols . . . . . 179  
`\MT@familyalias`: define alias font name as an alternative, not as a replacement . . . . . 59  
`\MT@get@basefamily`: also remove 'w' (swash capitals) . . . . . 87  
`\MT@get@highlevel`: check whether defaults have changed . . . . . 104  
`\MT@get@inh@list`: fix: set inheritance list `\globally` to `\@empty` . . . . . 90  
`\MT@get@listname@`: alternatively check for alias font name . . . . . 88  
`\MT@get@size`: additional magic to catch some errors hijack `\set@fontsize` instead of `\setfontsize` 106  
`\MT@loop`: fix: new macro, used instead of `\loop` . . 49  
`\MT@maybe@do`: also check for alias font name . . . . 59  
`\MT@permute@@@@`: more sanity checks for `\SetProtrusion` and `\SetExpansion` . . . . . 120  
`\MT@setupfont`: also search for alias font file . . . . 57  
 fix: call `\@enc@update` if necessary . . . . . 57

### 2004/10/27 **Version 1.3**

General: fix: specifying load option does no longer require to give a name, too . . . . . 114  
 Font aliases: declare `aer`, `zer` and `hfor` as aliases of `cmr` . . . . . 141  
`\MT@fix@catcode`: check some category codes (compatibility with german) . . . . . 35  
`\MT@load@list`: check whether list exists . . . . . 86

### 2004/11/12 **Version 1.4**

General: check for `pdfcpot` . . . . . 54  
 don't use scratch registers in global definitions . . use `\pickup@font` instead of `\define@newfont` as the hook for `\MT@setupfont` . . . . . 98  
 use one instead of five counters . . . . . 50  
 Protrusion: tweak quote characters for `cmr` variants (OT1, T1, lmr) . . . . . 156  
`\microtypesetup`: fix: set the correct levels, and remember them; warning when enabling an option disabled in package options . . . . . 128  
`\SetExpansion`: fix: specifying extra options does no longer require to give a name, too . . . . . 111

### 2004/11/17 **Version 1.4a**

General: new option: `final` . . . . . 125  
`\MT@cfg@catcodes`: fix: reset some more catcodes when reading files (reported by *Michael Hoppe*) 87

2004/11/26 **Version 1.4b**

General: fix: set catcodes before reading global configuration file (reported by <i>Christoph Bier</i> ) . . .	127	form abcz (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) . . . . .	150	<code>\MT@ifdimen</code> : don't set <code>\MT@count</code> globally (save stack problem) . . . . .	46
slanted like italics . . . . .	159	<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> ) . . . . .	125	<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> ) . . . . .	123	<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 . . . . .	63
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> . . .	144	<code>\MT@set@pr@codes</code> : adjust protrusion factors before setting the inheriting characters . . . . .	61
Protrusion: settings for Bitstream Charter . . . . .	151	<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 . . . .	132
<code>\MT@cfg@catcodes</code> : reset catcode of '=' (compatibility with Turkish babel) . . . . .	87	disable automatic expansion for DVI output . . .	133
<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 . . . . .	126	tune CMR math letters (OML encoding) . . . . .	180
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 . . . . .	125	<code>\MT@get@inh@list</code> : correct message if selected is false . . . . .	89
restructure dtx file . . . . .	140	<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 . . . .	61
test whether numeric options receive a number	125	<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 . . .	159	<code>\MT@vinfo</code> : new macro instead of <code>\ifMT@verbose</code> . .	36
improve settings for numbers (pointed out by <i>Peter Muthesius</i> ) . . . . .	153		

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 . . . . .	154
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 . . . . .	127
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 . . . . .	131		
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 . . . . .	95
<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> . . . . .	134
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 . . . . .	90	<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 . . . . .	90
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> . . . . .	141	<code>\MT@in@rlist</code> : made recursive . . . . .	48
Font sets: add U encoding to <code>allmath</code> . . . . .	140	<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> ) . . . . .	144	<code>\MT@is@letter</code> : warning for non-ASCII characters . . . . .	93
Protrusion: add LY1 characters for Times . . . . .	159	<code>\MT@ledmac@setup</code> : character protrusion with <code>ledmac</code> . . . . .	52
settings for AMS math fonts . . . . .	183	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\@for</code> . . . . .	47
verified settings for slanted Computer Modern Roman . . . . .	168	<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 . . . . .	96	restore <code>csquotes</code> ’s active characters . . . . .	54
<code>\MT@declare@sets</code> : warning when redefining a set . . . . .	103	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> . . . . .	141
disable microtype setup inside <code>hyperref</code> ’s <code>\pdfstringdef</code> (reported by <i>Hàn Thê Thành</i> ) . . . . .	55	Font sets: add OT4 encoding to text sets . . . . .	140
fix: use <code>true</code> as the default value . . . . .	123	add T5 encoding to text sets . . . . .	140

Inheritance: add list for OT4	145	<code>\MT@exp@two@n</code> : new macros: less <code>\expandafters</code>	44
add list for T5 (requested by <i>Hàn Thê Thành</i> )	146	<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	154	<code>\MT@is@active</code> : redone: use <code>\set@display@protect</code>	94
settings for OT4 encoding (Computer Modern Roman, Palatino, Times)	150	<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	150	<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	110

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>	142	<code>\MT@in@clist</code> : fix	48
Protrusion: fix: remove ‘_’ from OT1 encoding	155	<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	150	<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 L <sup>A</sup> T <sub>E</sub> X3	47
<code>\MT@check@font@cx</code> : optimise context-sensitive setup	101	<code>\MT@permute@#@#@#@#</code> : don’t define permutations for unused encodings	119
<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	151	<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	140	tweak AMS settings	183
add QX encoding to text sets	140	<code>\DeclareCharacterInheritance</code> : fix: empty context	117
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i> )	146	<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> )	157	<code>\MT@get@ex@opt</code> : fix: evaluate preset	70
settings for Euro symbols (Adobe, ITC, marvosym)	191	<code>\MT@get@font@dimen</code> : warning for zero <code>fontdimen</code>	65
		<code>\MT@get@opt</code> : optimise: don’t reset when preset op-	

tion is set	66	<code>\SetProtrusion</code> : (et al.) optimise: unify keys for mandatory argument	110
set list name before presetting	66	(et al.) split keys of optional and mandatory argument	110
<code>\MT@is@active</code> : support for Unicode (inputenc/utf8)	94		
<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when tex4ht is loaded (reported by <i>Peter Dyballa</i> )	54		

2006/07/28 **Version 1.9e**

General: fix: default value for <code>activate: true</code>	123	settings for Euler Roman font	187
Documentation: add hint about unknown encodings	26	<code>\DeclareCharacterInheritance</code> : new key ‘inputenc’ to set the input encoding	117
include LPL	243	<code>\MT@rem@from@clist</code> : model after <code>\@removeelement</code>	48
Font aliases: declare <code>zeur</code> and <code>zeus</code> ( <code>eulervm</code> ) as aliases of <code>eur</code> resp. <code>eus</code> ( <code>euler</code> )	142	<code>\MT@setup@</code> : empty <code>\MT@setup@</code> after use (compatibility with the <code>combine</code> class)	51
Inheritance: adapt to <code>marvosym</code> ’s changed encoding	148	<code>\pickup@font</code> : no tracing with <code>trace</code> package	99
Protrusion: complete settings for Euler Fraktur and Script fonts	190	<code>\SetExpansion</code> : new key: <code>inputenc</code>	111
fix: forgotten comma in <code>mt-mvs.cfg</code> ; adapt to <code>marvosym</code> ’s changed encoding	191	<code>\SetProtrusion</code> : (et al.) new key: <code>inputenc</code>	110

2006/09/09 **Version 1.9f**

Protrusion: fix: <code>euler-vm</code> did not load <code>euler</code> settings	187	<code>\MT@reset@context</code> : only reset context if it has actually been changed	102
<code>\MT@curr@list@name</code> : fix: <code>\MessageBreak</code> must not be expanded	96	<code>\MT@set@inh@list</code> : fix: forgotten comma in the features list	118
<code>\MT@gdef@n</code> : new macros: global variants	44	<code>\MT@set@named@keys</code> : new macro: set name first, simplify parsing of optional argument	112
<code>\MT@get@inh@list</code> : fix: input encoding must be set after the inheritance list has been parsed	90	<code>\SetProtrusion</code> : (et al.) set catcodes before parsing optional argument	110
<code>\MT@glet</code> : 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: <code>smallcaps</code>	140
compatibility with <code>soul</code> : register <code>\textls</code> and <code>\lststyle</code>	55	<code>\DeclareMicrotypeBabelHook</code> : new command: interaction with <code>babel</code>	110
new option: <code>babel</code> , by default <code>false</code> (language-dependent setup suggested by <i>Ulrich Dirr</i> )	123	<code>\lststyle</code> : fix: font switches don’t pose a problem anymore	77
new option: <code>letterspace</code> , by default <code>100</code>	125	fix: <code>letterspacing</code> commands may be nested	77
new package <code>letterspace</code> : a stripped-down version, containing the <code>letterspacing</code> commands only	1	new command: <code>letterspacing</code>	77
option ‘ <code>babel</code> ’: fix: switch off French <code>babel</code> ’s short-hands properly (reported by <i>Daniel Flipo</i> )	138	totally redone, using the new <code>\letterspacefont</code>	77
option ‘ <code>babel</code> ’: switch off Turkish <code>babel</code> ’s short-hands	138	<code>\MT@declare@sets</code> : fix: empty size list when redefining set	103
option ‘ <code>unit</code> ’, <code>\SetProtrusion</code> : deprecate value ‘ <code>relative</code> ’ completely	116	<code>\MT@is@symbol</code> : made even more robust	94
Documentation: add hint about how to increase <code>font_max</code> and <code>font_mem_size</code>	27	<code>\MT@load@inputenc</code> : sanitise catcodes before loading input encoding (problem with listings)	67
add hint about warning when tracking and expansion is applied to a font	28	<code>\MT@pdf@tex@no</code> : case 6: <code>pdfTeX 1.40</code>	40
add remark about ‘ <code>draft</code> ’ option disabling microtype (noted by <i>Michalis Miatidis</i> )	9	<code>\MT@setup@noligatures</code> : maybe disable <code>\MT@noligatures</code> after the preamble	137
qualify hint about web documents with regard to older <code>pdfTeX</code> versions	25	<code>\MT@split@name</code> : adjust to possible <code>letterspacing</code>	59
qualify hints about expansion error messages with regard to older <code>pdfTeX</code> versions	27	<code>\SetExtraKerning</code> : new command: additional kerning	112
Font sets: new: <code>footnotesize</code> and <code>scriptsize</code>	140	<code>\SetExtraSpacing</code> : new command: adjustment of interword spacing	112
		<code>\SetTracking</code> : new command: tracking	111
		<code>\textls</code> : new command: <code>letterspacing</code>	82
		starred version: remove spaces around text	82
		<code>\tracingmicrotypeinpdf</code> : new debug method: mark all fonts with PDF annotations	37

2007/01/21 **Version 2.1**

General: compatibility with pinyin: disable microtype in <code>\py@macron</code> (reported by <i>Sven Naumann</i> )	56	<code>\MT@get@ls@basefont</code> : redone: use <code>\pdfmatch</code> to make it bullet-proof	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	95
new option: <code>copyfonts</code>	124	<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> )	142	<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> ]	142	possibility to customise interword spacing	75
Font sets: default set for tracking: <code>smallcaps</code>	141	<code>\MT@setup@expansion</code> : warning if stretch or shrink aren't multiples of <code>step</code>	134
Inheritance: remove <code>'-'</code> → <code>'127'</code>	144	<code>\MT@setupfont</code> : don't call <code>\@enc@update</code> anymore	57
Protrusion: settings for Bitstream Letter Gothic	151	only add features that are available with the respective pdfTeX	57
Spacing: add sample	192	<code>\MT@setupfont@hook</code> : restore percent character if Galician <code>babel</code> is loaded	54
Tracking: add ligatures that are to be disabled	148	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letterspaced fonts	62
<code>\DeclareMicrotypeVariants</code> : new command	108	<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	111
<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> )	111
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	78	third argument may be empty	111
<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	138	<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> )	125	<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 . . . . .	111
possibility to customise outer kerning (suggested by <i>Stephan Hennig</i> ) . . . . .	75	<code>\MT@setup@tracking</code> : enable protrusion when tracking is enabled . . . . .	135
		<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 . . . . .	136
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 . . . . .	156	<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 . . . . .	84	<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> ) . . . . .	146
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> ) . . . . .	78
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> . . . . .	77	fix: don't adjust in math mode (reported by <i>Christoph Bier</i> ) . . . . .	81
make invalid in math mode . . . . .	77	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 . . . . .	127	<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> ) . . . . .	193
Expansion: settings for T2A encoding (contributed by <i>Karl Karlsson</i> ) . . . . .	149	<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 . . . . .	140	<code>\MT@setup@</code> : make space-unaware (requested by <i>Marcin Borkowski</i> ) . . . . .	51
add T2A encoding . . . . .	140	<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> ) . . . . .	157	<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*) . . . . . 157

2013/03/13 **Version 2.5**

General: allow contexts for LuaTeX . . . . . 114

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` . . . . . 216

Font aliases: declare `lmsy` and `lmm` as aliases of `cmsy` resp. `cmm` (reported by *Jonas Hogstrom*) . . . . . 141

declare `zgmx` etc. (`garamondx`) as aliases of `ugm` . . . . . 142

declare Latin Modern Roman (OpenType version) as alias of `lmr` when `fontspec` is loaded . . . . . 141

declare TeX Gyre Pagella, Asana Math, Palatino LT Std, and Palatino as aliases of Palatino Linotype (OpenType version) . . . . . 142

Font sets: add EU1 and EU2 encodings . . . . . 140

Inheritance: add rudimentary list for EU1 and EU2 . . . . . 147

Protrusion: add default lists for EU1 and EU2 . . . . . 155

improvements to Computer Modern Roman italics (contributed by *Hendrik Vogt*) . . . . . 160

Tracking: add EU2 encoding to default list . . . . . 148

`\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*) . . . . . 101

`\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@l@s@set@l@s`: 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*) . . . . . 61

`\MT@setup@spacing`: warning with `ragged2e` (reported by *Steffen Hoffmann*) . . . . . 135

`\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 `luatexextra` (contributed by *Élie Roux*) . . . . . 43

Documentation: add notes on typesetting the documentation . . . . . 28

include OpenType configuration files . . . . . 197

`\MT@afteraftergroup`: fix: get outer kerning and spacing of nested `letterspacing right` . . . . . 76

`\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@l@s@outer@k`: add marker for tightly nested `letterspacing` . . . . . 84

`\MT@set@tr@codes`: fix: load font for `fontspec` . . . . . 75

`\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` . . . . . 197

suggest to use `etoolbox` to patch `\verbatim` . . . . . 26

Font sets: add TU encoding (notified by *Will*) . . . . . 26

<i>Robertson</i> .....	140	<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> )	140	<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> ) .....	140	<code>\MT@luatex@no</code> : update for LuaTeX 0.85 (renamed primitives) .....	41
Inheritance: add TU encoding .....	147	<code>\MT@noligatures@</code> : use <code>luaotfload</code> function to keep/inhibit ligatures .....	85
Protrusion: add TU encoding to lists .....	155	<code>\MT@orig@pickupfont</code> : (in)compatibility with <code>luatexja</code> : disable unknown slots warnings (reported by <i>Max</i> ) .....	98
Tracking: add TU encoding to default list .....	148	(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 .....	98
<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 .....	108	<code>\MT@xspace</code> : fix outer spacing problem with (not only) algorithm (reported by <i>Henning</i> and <i>Ronnie Marksch</i> ) .....	82
<code>\sstyle</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	103		
<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</code> ABD	53		
<code>\MT@get@slot@</code> : fix: could fail with X <sub>Y</sub> 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 L <sup>A</sup> TeX .....	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 L <sup>A</sup> TeX 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 'file:font' 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 .....	120
Font aliases: declare aliases for <code>newpx</code> .....	142	<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older LuaTeX versions .....	69
declare aliases for <code>newtx</code> .....	142	<code>\MT@setup@expansion</code> : don't disable automatic expansion for DVI output with LuaTeX .....	133
declare aliases for <code>tempora</code> .....	142	<code>\MT@tikz@setup</code> : compatibility with <code>tikz</code> (again) ..	53
declare aliases for <code>XCharter</code> .....	142	<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 L <sup>A</sup> TeX format (reported by <i>Ulrike Fischer</i> )	141		
Protrusion: automatically choose correct names for Charis SIL small caps (reported by <i>'lcomdata'</i> )	209		
<code>\sstyle</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		

**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>	<code>DVIoutput</code> .....	<b>8</b>	<code>letterspace</code> .....	<b>8</b>
	<code>activate</code> .....	<b>6</b>	<code>protrusion</code> .....	<b>6</b>
	<code>auto</code> .....	<b>7</b>	<code>selected</code> .....	<b>8</b>
	<code>babel</code> .....	<b>9</b>	<code>shrink</code> .....	<b>8</b>
	<code>config</code> .....	<b>9</b>	<code>spacing</code> .....	<b>6</b>
	<code>draft</code> .....	<b>9</b>	<code>step</code> .....	<b>8</b>
	<code>expansion</code> .....	<b>6</b>	<code>stretch</code> .....	<b>8</b>
	<code>factor</code> .....	<b>7</b>	<code>tracking</code> .....	<b>6</b>
	<code>final</code> .....	<b>9</b>	<code>unit</code> .....	<b>7</b>
	<code>kerning</code> .....	<b>6</b>	<code>verbose</code> .....	<b>9</b>

- Commands**
- `\DeclareCharacterInheritance` ..... 20
  - `\DeclareMicrotypeAlias` ..... 21
  - `\DeclareMicrotypeBabelHook` ..... 23
  - `\DeclareMicrotypeSet*` ..... 10
  - `\DeclareMicrotypeSet` ..... 10
  - `\DeclareMicrotypeSetDefault` ..... 12
  - `\DeclareMicrotypeVariants*` ..... 20
  - `\DeclareMicrotypeVariants` ..... 20
  - `\DisableLigatures` ..... 24
  - `\LoadMicrotypeFile` ..... 22
  - `\SetExpansion` ..... 14
  - `\SetExtraKerning` ..... 18
  - `\SetExtraSpacing` ..... 19
  - `\SetProtrusion` ..... 13
  - `\SetTracking` ..... 15
  - `\UseMicrotypeSet` ..... 12
  - `\slig` ..... 24
  - `\sstyle` ..... 23
  - `\microtypecontext` ..... 22
  - `\microtypesetup` ..... 9
  - `\textls*` ..... 23
  - `\textls` ..... 23
  - `\textmicrotypecontext` ..... 22
- A**
- `aoposter` (package) ..... 106
  - `activate` (option) ..... 6, 123, 230
  - `\add@accent` ..... 2901
  - `\adjustspacing` ..... 4260
  - `ae` (package) ..... 21, 141
- B**
- `babel` (option) ..... 9, 23, 26, 31, 123, 230
  - `babel` (package) ..... 2, 3, 5, 17, 23, 24, 26, 53,
- C**
- `chemsym` (package) ..... 227
  - `chmath` (package) ..... 21, 142, 231
  - `CJK` (package) ..... 27, 56, 98, 231, 232, 234
  - `CJKutF8` (package) ..... 99, 232
  - `cm-super` (package) ..... 8
  - `color` (package) ..... 9, 132
  - `combine` (package) ..... 139, 230
- D**
- `\DeclareCharacterInheritance` ..... 20, 41, 3706
  - `\DeclareMicrotypeAlias` ..... 21, 43, 71, 3300
  - `\DeclareMicrotypeBabelHook` ..... 23, 45, 64, 3366
  - `\DeclareMicrotypeSet` ..... 10, 32, 57, 59, 72, 3018
  - `\DeclareMicrotypeSet*` ..... 10, 3018
  - `\DeclareMicrotypeSetDefault` ..... 12, 34, 61, 3255
  - `\DeclareMicrotypeVariants` ..... 20, 42, 58, 63, 3285
  - `\DeclareMicrotypeVariants*` ..... 20
- E**
- `eco` (package) ..... 21, 141
  - `\efcode` ..... 1530, 1556, 1557, 1597, 1599
  - `eledmac` (package) ..... 52, 233
  - `eplain` (package) ..... 24, 31, 34, 38, 52, 131, 231
  - `e-TeX` (engine) ..... 32, 38, 40, 44, 45, 50, 64, 81, 216, 227–229, 231
  - `etoolbox` (package) ..... 26, 233
  - `eucal` (package) ..... 188
  - `eufrak` (package) ..... 190
- F**
- `\f@family` ..... 2852, 2853
  - `\f@size` ..... 1897, 2845, 2847, 2849, 2856, 2899
  - `factor` (option) ..... 7, 13, 32, 125, 126, 227
  - `fancyvrb` (package) ..... 26
  - `final` (option) ..... 9, 33, 123, 125, 226
  - `fix-cm` (package) ..... 59
  - `\fmtversion` ..... 355
  - `\font` ..... 971, 1799, 1944
  - `\font@name` ..... 111, 1017, 1019, 1021, 1023, 1033, 1034, 1789, 1797, 1801, 1805, 1843, 1849, 1854, 1857, 1930, 1944, 1956, 1958, 1963, 1965, 1968, 1971, 1974, 2009, 2064, 2201, 2203, 2838, 2846, 2855, 2860, 2880, 2882, 2884, 2900, 2913, 2914, 2931, 2936
  - `\fontcharwd` ..... 1273, 1277, 1283
  - `\fontdimen` ..... 1196, 1203, 1308, 1313, 1412, 1415, 1818, 1819, 1849, 1956, 2002, 2003, 2010, 2011, 2017, 2021, 2028, 2038, 2042, 2045, 2064, 2214, 2217
  - `fontinst` (package) ..... 194, 216, 218
  - `fontinstallationguide` (package) ..... 173
  - `\fontname` ..... 1797, 1801
  - `fontspec` (package) ..... 6, 25, 27, 30, 43, 49, 53, 59, 77, 95, 104, 113, 141, 147, 233, 234
  - `fourier` (package) ..... 62
  - `french` (package) ..... 228
  - `frenchpro` (package) ..... 95, 227, 231
- G**
- `garamondx` (package) ..... 21, 142, 233
  - `german` (package) ..... 35, 226
  - `\glb@currsiz` ..... 2962
  - `algorithm` (package) ..... 234
  - `amsmath` (package) ..... 233
  - `amssymb` (package) ..... 183
  - `article` (package) ..... 22
  - `auto` (option) ..... 7, 123, 54, 87, 110, 138, 139, 143, 227, 228, 230–232
  - `config` (option) ..... 9, 20, 32, 127, 228
  - `contour` (package) ..... 132
  - `\copyfont` ..... 1012
  - `copyfonts` (option) ..... 57, 123, 124, 231
  - `crop` (package) ..... 132
  - `csquotes` (package) ..... 29, 53, 54, 154, 228
  - `\curr@fontshape` ..... 2845, 2847, 2849, 2856, 2899
  - `deferssetup` (option) ..... 54, 123, 124, 128, 229
  - `\define@newfont` ..... 2838, 2846, 2850, 2860
  - `\DisableLigatures` ..... 24, 40, 62, 3339
  - `\do@subst@correction` ..... 2898
  - `docstrip` (package) ..... 34
  - `draft` (option) ..... 9, 123, 125, 230
  - `dsfont` (package) ..... 62
  - `DVIOutput` (option) ..... 8, 9, 123, 124, 227, 233
  - `euler` (package) ..... 96, 187, 230
  - `eulervm` (package) ..... 21, 142, 187, 230
  - `euroitc` (package) ..... 191
  - `europs` (package) ..... 191
  - `eurosans` (package) ..... 191
  - `\expandglyphsinfont` ..... 1524
  - `expansion` (option) ..... 6, 11, 123, 132
  - `exscale` (package) ..... 234
  - `\glb@settings` ..... 1933
  - `graphics` (package) ..... 9, 132

- H** hfoldsty (package) . . . . . 21, 141
- I** IEEEtran (package) . . . . . 98  
 \iffontchar . . . . . 1318, 2774, 2786, 2787  
 \ifMT@auto . . . . . 297, 4452, 4503  
 \ifMT@babel . . . . . 297, 4579, 4707  
 \ifMT@do . . . . . 1066, 1129, 1765, 2248  
 \ifMT@document . . . . . 338, 3817  
 \ifMT@draft . . . . . 297, 4357  
 \ifMT@expansion . . . . . 297, 4415, 4500, 4536  
 \ifMT@fontspec . . . . .  
 . 829, 842, 853, 888, 1047, 3085, 3490, 3516  
 \ifMT@if@ . . . . . 296,  
 858, 865, 895, 927, 2927, 3924, 4727, 4730  
 \ifMT@inannot . . . . . 99  
 \ifMT@inlist@ . . . 614, 659, 756, 786, 989, 1096,  
 1113, 1122, 1140, 1158, 1763, 2352, 2363,  
 2412, 2461, 2914, 2921, 2933, 2988, 3000, 3322  
 \ifMT@kerning . . . . . 297, 4593, 4720
- J** jurabib (package) . . . . . 98
- K** kerning (option) . . . . . 6, 11, 31, 123  
 keyval (package) . . . . . 41, 63, 143, 231  
 \knaccode . . . . . 1714, 1715, 1725, 1728, 1734
- L** ledmac (package) . . . . . 29, 32, 52, 98, 228  
 ledpar (package) . . . . . 29  
 \leftmarginkern . . . . . 799  
 letterspace (option) 8, 17, 23, 24, 31, 45, 74, 125, 230  
 letterspace (package) . . . . . 1,  
 24, 31, 34, 36, 38, 53, 75, 76, 78, 230–232, 234  
 \letterspacefont . . . . . 1805  
 lineno (package) . . . . . 52  
 listings (package) . . . . . 26, 55, 67, 87, 229, 230, 233  
 lmodern (package) . . . . . 8, 156  
 \LoadMicrotypeFile . . . . . 22, 44, 74, 3318, 4362  
 \lpcode . . . . . 1210, 1246, 1247, 1450, 1453, 1832
- M** marvosym (package) . . . 21, 31, 34, 148, 191, 229, 230  
 mathastext (package) . . . . . 54, 233  
 mathdesign (package) . . . . . 21, 142, 229  
 memoir (package) . . . . . 22, 98  
 \Microtype@Hook . . . . . 127, 4191  
 \microtypecontext . . . . .  
 . . 22, 47, 916, 2957, 2968, 4364, 4690, 4693  
 microtypecontext (environment) . . . . . 22  
 \microtypesetup . . . . . 9, 46, 4195, 4363  
 MiKTeX (distribution) . . . . . 28  
 miniltx (package) . . . . . 24, 31, 34, 38, 52, 231  
 minimal (package) . . . . . 51, 56, 234  
 \MT@font . . . . . 397, 992,  
 1015, 1198, 1309, 1322, 1777, 2415, 2487, 2885  
 \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  
 hyperref (package) . . . . . 9, 54, 55, 98, 132, 228, 231  
 \ifMT@nofamily . . . . . 1168, 1175  
 \ifMT@noligatures . . . . . 297, 4673  
 \ifMT@nonselected . . . . . 1506, 2474, 2482  
 \ifMT@norest . . . . . 2543, 2560, 2573, 2587, 2648  
 \ifMT@opt@auto . . . . . 3935, 4456  
 \ifMT@opt@DVI . . . . . 3935, 4382  
 \ifMT@opt@expansion . . . . . 3935, 4411, 4537  
 \ifMT@protrusion . . . . . 297, 793, 816, 4394, 4552  
 \ifMT@selected . . . . . 297, 4486, 4506  
 \ifMT@spacing . . . . . 297, 4561, 4578  
 \ifMT@tracking . . . . . 297, 4548, 4608  
 \ifMT@xunicode . . . . . 829, 2731, 2778  
 ifpdf (package) . . . . . 9, 99, 132  
 \iftracingmicrotypeinpdfall . . . . . 105  
 \ignoreligaturesinfont . . . . . 2252  
 inputenc (package) . . . . .  
 . . 14, 26, 31, 55, 67, 90, 94, 96, 228, 230, 231  
 \nbcrcode . . . . . 1707, 1708, 1724, 1727, 1733  
 \nbscode . . . . . 1628, 1629, 1652, 1656, 1662  
 \lslig . . . . . 24, 54, 1949, 1952  
 \lststyle . . . . . 23, 51, 918, 943, 948,  
 1931, 2140, 2146, 2151, 2154, 2164, 2168, 4365  
 luaotfload (package) . . . . .  
 . . . . . 27, 30, 43, 49, 53, 75, 85, 92, 233, 234  
 LuaTeX (engine) . . . . . 1,  
 4, 6–8, 12–16, 20, 23–31, 34, 37–41, 43,  
 45, 49, 53, 54, 69, 74, 75, 82, 84, 85, 91,  
 114, 124, 130, 136, 141, 143, 147, 155, 231–234  
 luatexbase (package) . . . . . 43, 233, 234  
 luatexja (package) . . . . . 27, 98, 233, 234  
 luatextra (package) . . . . . 233  
 \MT@abbr@tr@c . . . . . 732  
 \MT@active@features . . . . .  
 1025, 2915, 2918, 2930, 2941, 2977, 2987,  
 3347, 4395, 4501, 4549, 4562, 4594, 4682, 4683  
 \MT@addto@annot . . . . . 99  
 \MT@addto@setup . . . . . 777,  
 778, 869, 1425, 2829, 2957, 2958, 3577,  
 3595, 4049, 4196, 4356, 4639, 4681, 4706, 4739  
 \MT@adjustspacing . . . . . 4256, 4257, 4502  
 \MT@afteraftergroup . . . . . 1861, 1869, 1878  
 \MT@auto . . . . . 1516, 4451, 4467, 4478, 4482  
 \MT@auto@ . . . . . 1516, 1526, 1570  
 \MT@autofalse . . . . . 299, 4464, 4477  
 \MT@autotruer . . . . . 299, 4151  
 \MT@babelfalse . . . . . 306  
 \MT@babeltrue . . . . . 306  
 \MT@begin@catcodes . . . . . 2353, 2354, 2394,  
 3019, 3225, 3256, 3287, 3301, 3326, 3342,  
 3376, 3392, 3414, 3435, 3449, 3712, 4166, 4167  
 \MT@cat . . . . . 1421, 2820, 2822  
 \MT@cfg@catcodes . . . . . 929, 1443, 2379, 2396  
 \MT@char . . . . . 1235,  
 1236, 1246, 1247, 1252, 1253, 1256, 1258,  
 1273, 1274, 1277, 1450–1454, 1556, 1557,  
 1559, 1560, 1597–1599, 1628, 1629, 1635,  
 1636, 1642, 1643, 1646, 1647, 1652–1657,  
 1707, 1708, 1714, 1715, 1718, 1719, 1724–  
 1728, 2260, 2261, 2263, 2516, 2532, 2540,

- 2542, 2552, 2554, 2562, 2563, 2568, 2572,  
 2573, 2578, 2580, 2586, 2588, 2591, 2595,  
 2599, 2603, 2715, 2718, 2720, 2769, 2774,  
 2776, 2779, 2781, 2787, 3762–3764, 3769–3771  
 \MT@char@ . . . . . 1279, 1280,  
 1283, 1318, 2516, 2522, 2527, 2532, 2581,  
 2652, 2654, 2660, 2661, 2663, 2676, 2677,  
 2680, 2681, 2684, 2685, 2689, 2691, 2719,  
 2738, 2741, 2745, 2751, 2758, 2760, 2777, 2794  
 \MT@charstring . . . . . 2529, 2718, 2723, 2750  
 \MT@charxstring . . . . . 2753, 2781  
 \MT@check@active@set . . . . .  
 . . . . . 4177, 4401, 4519, 4551, 4571, 4598  
 \MT@check@font . . . . . 988, 2911, 2979  
 \MT@check@font@cx . . . . . 2916, 2979  
 \MT@check@range . . . . . 3895, 3897  
 \MT@check@range@ . . . . . 3897, 3898  
 \MT@check@rlist . . . . . 3844, 3888  
 \MT@check@rlist@ . . . . . 3888, 3889  
 \MT@check@step . . . . . 4507  
 \MT@checklist@ . . . . . 1074, 1089, 2245  
 \MT@checklist@family . . . . . 1107  
 \MT@checklist@font . . . . . 1151  
 \MT@checklist@size . . . . . 1135  
 \MT@checksetup . . . . . 4207, 4218, 4225, 4244, 4280  
 \MT@clear@options . . . . . 161, 289, 767, 4140, 4349  
 \MT@clist@break . . . . . 586, 1102, 1130, 1146, 1160  
 \MT@clist@function . . . . . 586  
 \MT@cnt@encoding . . . . . 3783, 3791, 3792  
 \MT@cnt@family . . . . . 3789, 3798, 3799  
 \MT@cnt@series . . . . . 3796, 3805, 3806  
 \MT@cnt@shape . . . . . 3803, 3811, 3812  
 \MT@config@file 4155, 4164, 4165, 4169, 4170, 4173  
 \MT@context . . . . . 2437, 2449, 2468, 2495  
 \MT@copy@font . . . . . 975,  
 1010, 3573, 3579, 3591, 3598, 4058, 4060, 4385  
 \MT@copy@font@ . 1010, 3573, 3579, 3591, 3598, 4058  
 \MT@count . . . . . 503, 504,  
 716, 1265, 1273, 1275, 1277, 1281, 1283,  
 1286, 1292, 1297, 1298, 1302, 1313, 1344, 1406  
 \MT@curr@file . . . . . 2357, 2358, 2367, 2368,  
 3328, 3329, 3547, 3549, 3552, 3734, 4169, 4741  
 \MT@curr@list@name . . . 1311, 1347, 1437, 1566,  
 2257, 2338, 2506, 2789, 2797, 2802, 2810, 2816  
 \MT@curr@ls . . . . . 1840, 1953, 1956  
 \MT@curr@set@name . . . . . 3043,  
 3045–3047, 3049, 3052, 3053, 3058, 3063,  
 3067, 3068, 3098, 3102, 3131, 3161, 3167,  
 3173, 3177, 3178, 3429, 3505, 3526, 3541,  
 3560, 3632, 3636, 3643, 3646, 3648, 3654,  
 3657, 3660, 3672, 3685, 3693, 3701, 3735, 3737  
 \MT@declare@char@inh . . . . . 3717, 3722, 3732  
 \MT@declare@sets . . . . . 3026, 3031, 3044, 3349  
 \MT@DeclareMicrotypeAlias . . . . . 3302, 3304  
 \MT@DeclareMicrotypeSetDefault . . . . . 3257, 3259  
 \MT@DeclareSet . . . . . 3022, 3024, 3040  
 \MT@DeclareSetAndUseIt . . . . . 3021, 3039  
 \MT@DeclareVariants . . . . . 3289, 3290, 3292  
 \MT@def@bool@opt . . . . . 4012,  
 4025, 4026, 4030, 4044, 4056, 4066, 4078  
 \MT@def@n . . . . . 384, 3014, 3015  
 \MT@default@ex@set . . . . . 3274  
 \MT@default@kn@set . . . . . 3274  
 \MT@default@pr@set . . . . . 3274  
 \MT@default@sp@set . . . . . 3274  
 \MT@default@tr@set . . . . . 3274  
 \MT@define@code@key . . . . .  
 . . . . . 3473, 3533, 3535, 3536, 3744, 3746, 3747  
 \MT@define@code@key@family . . . . . 3484, 3534, 3745  
 \MT@define@code@key@font . . . . . 3510, 3538, 3749  
 \MT@define@code@key@size . . . . . 3498, 3537, 3748  
 \MT@define@opt@key . . . . . 3539, 3562–3565, 3697–3699  
 \MT@define@optionX . . . . . 4195, 4253, 4256  
 \MT@define@optionX@ . . . . . 4270, 4299, 4301, 4302  
 \MT@define@set@key@ . . . . . 3056, 3217–3220  
 \MT@define@set@key@font . . . . . 3165, 3222  
 \MT@define@set@key@size . . . . . 3091, 3221  
 \MT@detokenize@c . . . . . 427, 2524, 2716  
 \MT@detokenize@n . . . . . 427, 2771  
 \MT@dimen@six . . . . .  
 . . . . . 1195, 1266, 1302, 1344, 2037, 2040, 2210  
 \MT@dinfo . . . . . 87  
 \MT@dinfo@list . . . . .  
 1087, 1097, 1100, 1105, 1114, 1117, 1123,  
 1125, 1133, 1141, 1144, 1149, 1159, 1162, 1166  
 \MT@dinfo@nl . . . . . 87  
 \MT@DisableLigatures . . . . . 3339  
 \MT@do@font . . . . . 677, 1212, 1530, 1665, 1735, 1832  
 \MT@documentfalse . . . . . 338  
 \MT@documenttrue . . . . . 338, 4685  
 \MT@dofalse . . . . . 1066, 1078, 1101, 1118, 1145, 1163  
 \MT@dofont@function . . . . . 681, 698  
 \MT@dotrue 1066, 1069, 1098, 1115, 1124, 1142, 2240  
 \MT@draftfalse . . . . . 302, 4080  
 \MT@drafttrue . . . . . 302, 4082  
 \MT@edef@n . . . . . 386, 2473, 2499, 3004,  
 3479, 3493, 3549, 3552, 3555, 3733, 3875, 4112  
 \MT@encoding . . . . . 1044, 1155, 1176, 1178, 1181,  
 2432, 2444, 2524, 2716, 2770, 2801, 2808, 2816  
 \MT@end@catcodes . . . . . 2355,  
 2398, 3037, 3241, 3272, 3298, 3316, 3334,  
 3351, 3387, 3411, 3430, 3446, 3460, 3727, 4168  
 \MT@endinput . . . . . 165, 4135, 4144  
 \MT@engine . . . . . 173,  
 270, 274, 280, 286, 2141, 2145, 4136, 4142  
 \MT@engine@toold . . . . . 173, 271  
 \MT@error . . . . . 76, 759,  
 872, 2141, 2152, 2330, 2343, 3129, 3250,  
 3279, 3357, 3582, 3608, 3615, 4068, 4230,  
 4248, 4288, 4457, 4471, 4538, 4609, 4622, 4702  
 \MT@error@doesnt@work . . . . . 4605  
 \MT@ex@cc@name . . . . . 1498,  
 1500, 1572, 1573, 1583, 1589, 1590, 1602, 3395  
 \MT@ex@context . . . . . 1015, 2971, 3013  
 \MT@ex@doc@contexts . . . . . 3013  
 \MT@ex@factor . . . . . 307, 1517, 1576  
 \MT@ex@factor@ . 1517, 1532, 1537, 1546, 1547, 1570  
 \MT@ex@inh@name . . . . . 1558–1560  
 \MT@ex@level . . . . . 307, 4502, 4504  
 \MT@ex@max . . . . . 319, 1549, 1550  
 \MT@ex@min . . . . . 319, 1552, 1553  
 \MT@ex@setname . . . . . 3243  
 \MT@ex@split@val . . . . . 1544  
 \MT@exp@cs . . . . . 381, 384, 387, 390, 395, 1026,  
 1028, 1037, 1139, 1257, 1331, 1334, 1364,  
 1368, 1378, 1381, 1384, 1560, 1647, 1719,  
 1881–1883, 2460, 2934, 2942, 2943, 2947,  
 2990, 3001, 3097, 3504, 3524, 3771, 3846, 3893  
 \MT@exp@gcs . . . . . 381, 385, 389, 391, 396

- \MT@exp@one@n ..... 398, 432, 593,  
     979, 981, 1038, 1093, 1156, 1762, 2362,  
     2911, 2913, 2919, 2931, 2953, 2987, 2998,  
     3013, 3031, 3216, 3235, 3266, 3321, 3722, 4690  
 \MT@exp@two@c ..... 400, 418, 423, 433, 628,  
     978, 1019, 1021, 1023, 1032, 1800, 1930,  
     1965, 1968, 1971, 2523, 2528, 2717, 2779, 2780  
 \MT@exp@two@n ..... 402, 1111, 1120, 3062, 3172  
 \MT@expandfont ..... 1502, 1520, 1524  
 \MT@expansion ..... 994, 1489, 4529  
 \MT@expansionfalse ..... 298, 4412, 4496  
 \MT@expansiontrue ..... 298, 4150  
 \MT@extra@context ..... 3013, 3379, 3395, 3418,  
     3438, 3452, 3519, 3522, 3523, 3525, 3566,  
     3574, 3580, 3592, 3597, 3708, 3843, 3847,  
     3850, 3853, 3854, 3859, 3864, 3865, 3867, 3894  
 \MT@extra@inputenc ..... 3706, 3731, 3736, 3737  
 \MT@factor@default ..... 331, 4119, 4398  
 \MT@family ..... 979, 1044,  
     1112, 1155, 1176, 1178, 1181, 2433, 3312, 3313  
 \MT@familyalias .....  
     980, 981, 1055, 1119, 1121, 2443, 2445, 3314  
 \MT@feat ..... 1066, 1200, 1238, 1326, 1328,  
     1330, 1331, 1333, 1334, 1340, 1342, 1345,  
     1353–1357, 1359, 1361–1365, 1368, 1369,  
     1374, 1378, 1381, 1384, 1387–1389, 1398,  
     1399, 1404, 1422, 1934, 2328, 2330, 2333,  
     2334, 2338, 2340, 2343, 2469, 2471, 2473,  
     2477, 2481, 2485, 2486, 2489, 2497, 2499,  
     2500, 2502, 2509, 2513, 2790, 2791, 2820, 2822  
 \MT@features .....  
     752, 3013, 3026, 3216, 3230, 3261, 3544, 3717  
 \MT@features@long ..... 752, 755, 760, 2984, 3729  
 \MT@file@list ..... 2349,  
     2351, 2359, 2362, 2364, 2369, 2372, 3321, 3325  
 \MT@find@file ..... 979, 981, 2349  
 \MT@fix@catcode ..... 5  
 \MT@fix@font@set ..... 3081, 4369  
 \MT@font ..... 397, 712, 853, 889,  
     978, 1032, 1039, 1196, 1203, 1210, 1211,  
     1246, 1247, 1252, 1253, 1273, 1274, 1277,  
     1280, 1283, 1308, 1313, 1318, 1412, 1415,  
     1450, 1451, 1453, 1454, 1526, 1530, 1556,  
     1557, 1597, 1599, 1628, 1629, 1635, 1636,  
     1642, 1643, 1652–1654, 1656, 1657, 1662–  
     1664, 1707, 1708, 1714, 1715, 1724, 1725,  
     1727, 1728, 1733, 1734, 1762, 1766, 1838,  
     1839, 2249, 2539, 2594, 2824, 2880–2882,  
     2884, 2899, 2900, 2911, 2912, 2919, 2945, 2954  
 \MT@font@copy ..... 1015, 1023, 1024, 1032–1034  
 \MT@font@list ..... 2824, 2911–2914, 2978  
 \MT@font@orig ..... 1017  
 \MT@font@sets ..... 3066, 3081, 3176, 4369  
 \MT@fontspec@setup ..... 836, 884  
 \MT@fontspec@false ..... 832, 835  
 \MT@fontspec@true ..... 832, 834, 884  
 \MT@gdef@n ..... 384, 3370, 3386,  
     3410, 3445, 3459, 3648, 3672, 3685, 3739, 4185  
 \MT@get@axis ..... 3190–3193, 3203  
 \MT@get@basefamily ..... 2361, 2399  
 \MT@get@basefamily@ ..... 2405, 2408  
 \MT@get@char@unit ..... 1237, 1376, 1402, 1494  
 \MT@get@char@wd ..... 1271, 1302, 1379, 1402  
 \MT@get@config ..... 4155  
 \MT@get@ex@opt@ ..... 1493, 1511, 1570  
 \MT@get@ex@opt@ ..... 1578–1580, 1582, 1588  
 \MT@get@font ..... 3171, 3181  
 \MT@get@font@ ..... 3182, 3187, 3531  
 \MT@get@font@and@size ..... 3515, 3530  
 \MT@get@font@dimen ..... 1306, 1382  
 \MT@get@font@dimen@six 1184, 1195, 1612, 1691, 1778  
 \MT@get@highlevel ... 3061, 3071, 3205, 3478, 3489  
 \MT@get@inh@list ... 1187, 1496, 1615, 1694, 2495  
 \MT@get@listname ..... 2414, 2471, 2497  
 \MT@get@listname@ ..... 2414  
 \MT@get@ls@basefont ..... 1793, 1955, 1962  
 \MT@get@opt ..... 1185, 1351, 1613, 1692  
 \MT@get@range ..... 3095, 3105, 3502  
 \MT@get@size ..... 3112, 3117, 3126, 3145, 3201  
 \MT@get@slot ..... 1234, 2259, 2516, 3761, 3768  
 \MT@get@slot@ ..... 2533, 2537  
 \MT@get@space@unit .....  
     1376, 1478, 1626, 1633, 1640, 1705, 1712  
 \MT@get@tr@opt ..... 1780, 1900  
 \MT@get@tr@opt@ ..... 1918–1921, 1923  
 \MT@get@unit ..... 1384, 1392, 1911  
 \MT@get@unit@ ..... 1392  
 \MT@getkey ..... 4336, 4351  
 \MT@glet . 379, 391, 887, 1019, 1882, 1965, 1996,  
     2146, 2154, 2878, 2962, 2979–2982, 3314,  
     3573, 3591, 3786, 4058, 4060, 4653, 4659, 4745  
 \MT@glet@nc ..... 390, 1181, 2009, 2201, 2509,  
     2900, 2978, 3049, 3052, 3058, 3167, 3403,  
     3632, 3636, 3643, 3646, 3654, 3657, 3660, 3693  
 \MT@glet@nn ..... 395, 2997, 4182  
 \MT@if@false ..... 296,  
     854, 861, 891, 908, 2924, 3892, 4721, 4728  
 \MT@if@fontspec@font .....  
     680, 836, 1795, 1830, 1839, 2262, 2275  
 \MT@if@list@exists .....  
     1174, 1492, 1510, 1611, 1690, 1779, 2468  
 \MT@if@outer@next .....  
     2080, 2095, 2099, 2102, 2107, 2112–2115  
 \MT@if@true ..... 296,  
     855–857, 862–864, 892–894, 921, 925, 926,  
     2917, 3901, 3905, 3913, 3918, 4722–4726, 4729  
 \MT@ifdefined@c@T ..... 406, 858,  
     897, 1255, 1291, 1558, 1645, 1717, 1824,  
     2704, 3083, 3312, 3736, 3833, 4191, 4194, 4748  
 \MT@ifdefined@c@TF .....  
     406, 794, 1783, 1808, 1811, 1822,  
     1953, 2034, 2207, 2254, 2255, 2472, 2498, 3842  
 \MT@ifdefined@n@T .....  
     406, 783, 1090, 1108, 1136, 1152, 1256,  
     1387, 1423, 1559, 1583, 1646, 1718, 1903,  
     1905, 1924, 2451, 2819, 3046, 3307, 3398,  
     3556, 3792, 3799, 3806, 3812, 3843, 3853, 4336  
 \MT@ifdefined@n@TF .....  
     406, 1055, 1072, 1091, 1109, 1137,  
     1153, 1176, 1353, 1361, 1572, 1589, 1880,  
     2243, 2333, 2439, 2524, 3244, 3247, 3275,  
     3547, 3819, 3874, 3883, 3953, 4178, 4181, 4688  
 \MT@ifdim . . 526, 648, 649, 653, 654, 3128, 3137,  
     3899–3901, 3903, 3904, 3911–3913, 3916, 3917  
 \MT@ifdimen ... 488, 3156, 3635, 3645, 3659, 4128  
 \MT@ifempty ..... 439, 1210, 1211, 1244, 1250,  
     1467, 1468, 1625, 1632, 1639, 1662–1664,  
     1676–1678, 1680–1682, 1704, 1711, 1733,  
     1734, 1750, 1751, 2015, 2016, 2026, 2044,  
     2176, 2199, 2200, 2212, 2989, 3025, 3029,

- 3073, 3079, 3106, 3107, 3120, 3121, 3195,  
 3206, 3229, 3233, 3260, 3264, 3346, 3425,  
 3540, 3546, 3566, 3572, 3580, 3590, 3597,  
 3670, 3716, 3721, 3966, 3997, 4158, 4204, 4277  
 \MT@ifint ..... 452, 2683, 3426, 3671, 4111, 4121  
 \MT@ifstreq ..... 535, 1429,  
2329, 2470, 2485, 2739, 2742, 2750, 2759,  
2952, 3170, 3514, 3634, 3656, 3658, 3834,  
3854, 3925, 3969, 3971, 3974, 3977, 3999,  
4001, 4015, 4016, 4088, 4089, 4094, 4098,  
4127, 4206, 4213, 4217, 4224, 4279, 4285, 4581  
 \MT@in@clist ..... 614,  
755, 784, 1093, 1111, 1120, 1156, 1762,  
2351, 2362, 2911, 2913, 2919, 2931, 2987, 3321  
 \MT@in@rlist ..... 642, 1139, 2460  
 \MT@in@rlist@ ..... 642  
 \MT@in@rlist@ ..... 642  
 \MT@in@tlist ..... 630, 2411, 2998  
 \MT@in@tlist@ ..... 630  
 \MT@inannotfalse ..... 100  
 \MT@inannottrue ..... 100  
 \MT@increment ..... 716, 3791, 3798, 3805, 3811  
 \MT@info ..... 76, 91, 96, 4095, 4320, 4324  
 \MT@info@missing@char ..... 1286, 1316, 2590  
 \MT@info@nl ..... 76, 92, 93, 97, 795, 1177,  
1317, 4086, 4096, 4165, 4179, 4183, 4381,  
4385, 4397, 4404, 4503, 4530, 4550, 4555,  
4564, 4574, 4597, 4601, 4614, 4629, 4698, 4709  
 \MT@inh@do ..... 2508, 3750  
 \MT@inh@feat ..... 3706, 3716, 3719, 3730  
 \MT@inh@split ..... 3752, 3756  
 \MT@inlist@false ..... 614, 618, 631, 643, 2927  
 \MT@inlist@true ..... 614, 620, 638, 650, 655, 2927  
 \MT@is@active ..... 2523, 2696  
 \MT@is@char ..... 2528, 2717, 2723  
 \MT@is@charx ..... 2732, 2753, 2780  
 \MT@is@composite ..... 2526, 2767  
 \MT@is@feature ..... 754, 3030, 3234, 3265  
 \MT@is@letter ..... 2521, 2640, 2720, 2776  
 \MT@is@number ..... 2666, 2671  
 \MT@is@symbol ..... 2525, 2714  
 \MT@is@uni@comp ..... 2774, 2786, 2786  
 \MT@is@xchar ..... 2743, 2749  
 \MT@iterate ..... 664  
 \MT@kerning ..... 998, 1687, 4600  
 \MT@kerningfalse ..... 304  
 \MT@kerningtrue ..... 304  
 \MT@kn@cname ..... 1696, 1698, 1741, 3452  
 \MT@kn@context ..... 2972, 3013  
 \MT@kn@doc@contexts ..... 3013  
 \MT@kn@factor ..... 307  
 \MT@kn@factor@ ..... 1353  
 \MT@kn@inh@name ..... 1717–1719  
 \MT@kn@max ..... 319  
 \MT@kn@min ..... 319  
 \MT@kn@setname ..... 3243  
 \MT@kn@split@val ..... 1702  
 \MT@kn@unit ..... 313  
 \MT@kn@unit@ ..... 1361, 1744  
 \MT@led@unhbox@line ..... 795  
 \MT@ledmac@setup ..... 789, 878–880  
 \MT@let@cn 392, 395, 396, 839, 1056, 1191, 1500,  
1573, 1619, 1698, 1904, 1906, 1910, 2328,  
2340, 2465, 2469, 2502, 2880, 3149, 3560, 3735  
 \MT@let@nc .. 390, 1389, 2121, 2481, 2513, 2922,  
3876, 3881, 3975, 3978, 4220, 4227, 4612, 4627  
 \MT@let@nn .....  
395, 1354, 1359, 1362, 1374, 1590, 1593, 1925  
 \MT@letterspace .....  
 ... 334, 1783, 1904, 1914, 4662, 4663, 4665  
 \MT@letterspace@ ..... 1221, 1291,  
1292, 1783, 1784, 1787, 1790, 1798, 1805,  
1810, 1818, 1831–1833, 1840, 1849, 1893–  
1896, 2040, 2044, 2162, 2177–2180, 2213, 2216  
 \MT@letterspace@default ..... 334, 4663  
 \MT@listname ..... 2414,  
2462, 2465, 2472, 2473, 2475, 2477, 2498,  
2499, 2501, 2502, 2506, 2509, 3771, 3776  
 \MT@listname@count ..... 3543, 3548, 3550  
 \MT@load@inputenc ..... 1430, 1433, 1442  
 \MT@load@list ..... 1189, 1498, 1617, 1696, 2325  
 \MT@loop ..... 664, 672, 710, 3880  
 \MT@lower ..... 3099, 3105, 3505  
 \MT@ls@adjust ..... 1847, 2173  
 \MT@ls@adjust@ ..... 2157, 2181  
 \MT@ls@adjust@empty ..... 2159, 2173  
 \MT@ls@adjust@relax ..... 2160, 2173  
 \MT@ls@basefont ..... 1962, 1971, 1974, 1975  
 \MT@ls@fontspec@colon ..... 1889  
 \MT@ls@fontspec@font ..... 1800, 1889  
 \MT@ls@outer@k ..... 1850,  
1855, 1863, 2057, 2071, 2118, 2136, 2221  
 \MT@ls@set@ls ..... 2164, 2167, 2173  
 \MT@ls@too@large ..... 2180, 2183, 4665  
 \MT@lsfont ..... 1789, 1799, 1805,  
1818, 1819, 1828, 1832, 1833, 1838, 1930,  
1940, 1985, 1987, 2002, 2003, 2010, 2011,  
2017, 2021, 2028, 2038, 2042, 2045, 2214, 2217  
 \MT@ltx@pickupfont ..... 914, 958, 963, 2895, 2903  
 \MT@lua 251, 357, 476, 513, 558, 682, 2263, 2276, 2544  
 \MT@luatex@no ..... 249, 347  
 \MT@map@clist@ ..... 586  
 \MT@map@clist@c ..... 586,  
1025, 2258, 2918, 2930, 2941, 2977, 2984,  
3026, 3230, 3261, 3544, 3717, 3719, 3729, 4334  
 \MT@map@clist@n .....  
586, 1071, 2242, 3013, 3028, 3048, 3059,  
3093, 3168, 3216, 3232, 3263, 3293, 3368,  
3476, 3487, 3500, 3512, 3652, 3668, 3765,  
3961, 3964, 3993, 3995, 4024, 4106, 4202, 4275  
 \MT@map@tlist@ ..... 604  
 \MT@map@tlist@c ... 604, 633, 644, 1028, 1257,  
1560, 1647, 1719, 2418, 2947, 3844, 3893, 4369  
 \MT@map@tlist@n ..... 604, 2390, 2725, 3785  
 \MT@max@char ..... 2640, 2663  
 \MT@max@slot ..... 2640, 2689  
 \MT@maybe@do .. 1066, 1171, 1490, 1609, 1688, 1764  
 \MT@maybe@gobble@with@tikz ..... 843, 1879, 1933  
 \MT@maybe@rem@from@list ..... 2947, 2951  
 \MT@minus ..... 1798, 1893–1896  
 \MT@MT ..... 2, 76, 79, 82, 84,  
86, 91, 92, 94, 95, 285, 766, 873, 875, 3582,  
4138, 4159, 4320, 4324, 4382, 4585, 4702, 4703  
 \MT@MT@pickupfont ..... 960, 965, 2895, 2905  
 \MT@next@listname ..... 2421, 2430  
 \MT@next@listname@ ..... 2440, 2452, 2458  
 \MT@nl@ligatures ..... 2249, 3339  
 \MT@nl@setname ..... 2241, 3339  
 \MT@nofamilyfalse ..... 1168, 1173

- `\MT@nofamilytrue` . . . . . [1168](#), [2431](#)  
`\MT@noligatures` . . . . . [1002](#), [2237](#), [4674](#)  
`\MT@noligatures@` . . . . . [1985](#), [1987](#), [2249](#), [2252](#)  
`\MT@noligaturesfalse` . . . . . [301](#)  
`\MT@noligaturestrue` . . . . . [301](#), [3348](#)  
`\MT@nonselectedfalse` . . . . . [1506](#), [1521](#)  
`\MT@nonselectedtrue` . . . . . [1506](#), [1509](#)  
`\MT@noresetfalse` . . . . .  
. . . . . [2649](#), [2656](#), [2665](#), [2686](#), [2739](#), [2751](#), [2759](#)  
`\MT@noresettrue` . . . . . [2519](#), [2649](#), [2674](#)  
`\MT@old@cmd` . . . . . [66](#), [71–74](#)  
`\MT@opt@autofalse` . . . . . [3936](#)  
`\MT@opt@autotrue` . . . . . [3936](#), [4026](#)  
`\MT@opt@def@set` . . . . . [3952](#), [3980](#), [4005](#)  
`\MT@opt@DVIfalse` . . . . . [3937](#)  
`\MT@opt@DVITrue` . . . . . [3937](#), [4033](#), [4036](#)  
`\MT@opt@expansionfalse` . . . . . [3935](#)  
`\MT@opt@expansiontrue` . . . . . [3935](#)  
`\MT@options` . . . . . [3463](#)  
`\MT@optwarn@admissible` . . . . . [3938](#), [4017](#), [4098](#)  
`\MT@optwarn@nan` . . . . . [3942](#), [4113](#), [4123](#)  
`\MT@orig@add@accent` . . . . . [2901](#)  
`\MT@orig@foreign@language` . . . . . [4715](#), [4717](#)  
`\MT@orig@pickupfont` . . . . . [2830](#), [2897](#)  
`\MT@orig@py@macron` . . . . . [955](#), [959](#), [964](#)  
`\MT@orig@select@language` . . . . . [4710](#), [4712](#)  
`\MT@outer@kern` [1849](#), [1853](#), [1855](#), [1856](#), [1860](#), [1868](#),  
[1956](#), [1957](#), [1959](#), [2118](#), [2119](#), [2136](#), [2194](#), [2233](#)  
`\MT@outer@space` . . . . . [1843](#),  
[1844](#), [2000](#), [2053](#), [2055](#), [2056](#), [2069](#), [2070](#),  
[2087](#), [2088](#), [2103](#), [2104](#), [2108](#), [2109](#), [2134](#)  
`\MT@pdf@annot` . . . . . [99](#)  
`\MT@pdf@or@lua` . . . . . [178](#), [186](#), [283](#), [455](#), [4646](#)  
`\MT@pdftex@no` . . . . . [208](#), [346](#)  
`\MT@permute` [3385](#), [3409](#), [3423](#), [3444](#), [3458](#), [3742](#), [3781](#)  
`\MT@permute@` . . . . . [3781](#)  
`\MT@permute@@` . . . . . [3781](#)  
`\MT@permute@@@` . . . . . [3781](#)  
`\MT@permute@@@@` . . . . . [3810](#), [3815](#)  
`\MT@permute@@@@@` . . . . . [3823](#), [3825](#)  
`\MT@permute@define` . . . . . [3816](#), [3826–3828](#), [3872](#)  
`\MT@permute@reset` . . . . . [3785](#), [3878](#)  
`\MT@permutelist` . . . . . [3379](#), [3395](#),  
[3421](#), [3438](#), [3452](#), [3519](#), [3520](#), [3525](#), [3740](#),  
[3838](#), [3839](#), [3843](#), [3847](#), [3850](#), [3853–3855](#),  
[3857–3859](#), [3867](#), [3868](#), [3894](#), [3926](#), [3928](#), [3929](#)  
`\MT@pickupfont` . . . . . [2895](#)  
`\MT@plain` . . . . . [125](#), [146](#)  
`\MT@pr@cc@name` . . . . . [1189](#), [1191](#), [1458](#), [3379](#)  
`\MT@pr@context` . . . . . [1015](#), [2971](#), [3013](#)  
`\MT@pr@doc@contexts` . . . . . [3013](#)  
`\MT@pr@factor` . . . . . [307](#), [4122](#), [4398](#), [4399](#)  
`\MT@pr@factor@` . . . . . [1353](#)  
`\MT@pr@inh@name` . . . . . [1255](#), [1256](#), [1258](#)  
`\MT@pr@level` . . . . . [307](#), [4396](#), [4397](#)  
`\MT@pr@max` . . . . . [319](#)  
`\MT@pr@min` . . . . . [319](#)  
`\MT@pr@setname` . . . . . [3243](#)  
`\MT@pr@split@val` . . . . . [1242](#)  
`\MT@pr@unit` . . . . . [313](#), [4129](#), [4400](#)  
`\MT@pr@unit@` . . . . . [1361](#), [1461](#)  
`\MT@preset@aux` . . . . . [1463](#),  
[1465](#), [1467](#), [1468](#), [1471](#), [1746](#), [1748](#), [1750](#), [1751](#)  
`\MT@preset@aux@factor` [1463](#), [1471](#), [1676–1678](#), [1746](#)  
`\MT@preset@aux@space` . [1465](#), [1471](#), [1680–1682](#), [1748](#)  
`\MT@preset@ex` . . . . . [1584](#), [1601](#)  
`\MT@preset@kn` . . . . . [1739](#)  
`\MT@preset@kn@` . . . . . [1739](#)  
`\MT@preset@pr` . . . . . [1456](#)  
`\MT@preset@pr@` . . . . . [1456](#)  
`\MT@preset@sp` . . . . . [1669](#)  
`\MT@preset@sp@` . . . . . [1669](#)  
`\MT@ProcessOptionsWithKV` . . . . . [4328](#), [4352](#)  
`\MT@protrudechars` . . . . . [4253](#), [4257](#), [4396](#), [4552](#)  
`\MT@protrusion` . . . . . [993](#), [1170](#), [4403](#)  
`\MT@protrusionfalse` . . . . . [297](#)  
`\MT@protrusiontrue` . . . . . [297](#), [4145](#)  
`\MT@rbba@expansion` . . . . . [747](#)  
`\MT@rbba@kerning` . . . . . [747](#)  
`\MT@rbba@protrusion` . . . . . [747](#)  
`\MT@rbba@spacing` . . . . . [747](#)  
`\MT@rbba@tracking` . . . . . [747](#)  
`\MT@register@font` . . . . . [1006](#), [2912](#), [2980](#)  
`\MT@register@font@cx` . . . . . [2940](#), [2980](#)  
`\MT@register@subst@font` . . . . . [2886](#), [2913](#), [2981](#)  
`\MT@register@subst@font@cx` . . . . . [2929](#), [2981](#)  
`\MT@rem@from@clist` . . . . . [625](#), [1038](#), [2953](#), [3468](#)  
`\MT@rem@from@list` . . . . . [1028](#), [1036](#)  
`\MT@rem@last@space` . . . . . [427](#)  
`\MT@repeat` . . . . . [664](#), [674](#), [712](#), [3886](#)  
`\MT@requires@latex` [145](#), [162](#), [775](#), [833](#), [883](#), [910](#),  
[941](#), [1929](#), [2828](#), [3944](#), [4105](#), [4329](#), [4333](#), [4744](#)  
`\MT@requires@luatex` . . . . . [341](#), [985](#), [1012](#), [1219](#),  
[1524](#), [1534](#), [1758](#), [2252](#), [4029](#), [4258](#), [4272](#), [4546](#)  
`\MT@requires@pdftex` . . . . . [341](#),  
[454](#), [489](#), [536](#), [790](#), [974](#), [984](#), [997](#), [1001](#),  
[1013](#), [1218](#), [1263](#), [1289](#), [1533](#), [1608](#), [1687](#),  
[1757](#), [1982](#), [2238](#), [3340](#), [3570](#), [3684](#), [3692](#),  
[4055](#), [4149](#), [4271](#), [4423](#), [4454](#), [4481](#), [4545](#), [4672](#)  
`\MT@res@a` . . . . . [548](#), [550](#), [616](#), [623](#), [626](#), [628](#), [632](#), [637](#)  
`\MT@res@b` . . . . . [549](#), [550](#), [626–628](#), [636](#), [637](#)  
`\MT@reset@context` . . . . . [2961](#), [2965](#), [2969](#), [2996](#)  
`\MT@reset@context@` . . . . . [2969](#), [2996](#)  
`\MT@reset@ef@codes` . . . . . [1495](#), [1519](#), [1533](#), [1585](#)  
`\MT@reset@ef@codes@` . . . . . [1528](#), [1538](#), [1542](#)  
`\MT@reset@kn@codes` . . . . . [1693](#), [1700](#), [1730](#)  
`\MT@reset@kn@codes@` . . . . . [1730](#)  
`\MT@reset@pr@codes` . . . . . [1186](#), [1193](#), [1214](#)  
`\MT@reset@pr@codes@` . . . . . [1214](#)  
`\MT@reset@sp@codes` . . . . . [1614](#), [1621](#), [1659](#)  
`\MT@reset@sp@codes@` . . . . . [1659](#)  
`\MT@restore@catcodes` . . . . . [5](#), [7](#), [8](#), [163](#), [4750](#)  
`\MT@restore@p@h` . . . . . [826](#), [865](#), [927](#)  
`\MT@saved@setupfont` . . . . . [4317](#), [4325](#)  
`\MT@scale` . . . . . [723](#), [1266](#), [1327](#), [1341](#), [1344](#), [1547](#)  
`\MT@scale@factor` . . . . . [1268](#), [1299](#), [1325](#), [1473](#), [1603](#)  
`\MT@scale@to@em` . . . . . [1245](#), [1251](#), [1263](#), [1296](#), [1479](#),  
[1627](#), [1634](#), [1641](#), [1706](#), [1713](#), [1913](#), [2035](#), [2208](#)  
`\MT@scrubfeature` . . . . . [1048](#), [1059](#), [3491](#)  
`\MT@scrubfeatures` . . . . . [1059](#), [3086](#), [3517](#)  
`\MT@selectedfalse` . . . . . [300](#)  
`\MT@selectedtrue` . . . . . [300](#)  
`\MT@series` . . . . . [1044](#), [1155](#), [2434](#), [2446](#)  
`\MT@set@all@ex` . . . . . [1528](#), [1604](#)  
`\MT@set@all@kn` . . . . . [1730](#), [1752](#)  
`\MT@set@all@pr` . . . . . [1207](#), [1214](#), [1469](#)  
`\MT@set@all@sp` . . . . . [1659](#), [1684](#)  
`\MT@set@babel@context` . . . . . [4687](#), [4713](#), [4718](#), [4732](#)  
`\MT@set@codes` . . . . . [1192](#), [1225](#), [1501](#), [1620](#), [1699](#), [2341](#)

- \MT@set@curr@ls ..... 1840  
 \MT@set@curr@ok 1860, 1862, 1868, 1871, 2092, 2096  
 \MT@set@curr@os ..... 1842, 1870, 2092, 2096  
 \MT@set@default@set ..... 3261, 3266, 3274  
 \MT@set@ex@codes ..... 1523, 4487, 4489  
 \MT@set@ex@codes@n ..... 1506, 1523, 4489  
 \MT@set@ex@codes@s ..... 1491, 4487  
 \MT@set@ex@heirs ..... 1560, 1596  
 \MT@set@inh@list ..... 3713, 3715  
 \MT@set@inputenc 1188, 1420, 1497, 1616, 1695, 2507  
 \MT@set@inputenc@ ..... 1423, 1425  
 \MT@set@kn@codes ..... 1689  
 \MT@set@kn@heirs ..... 1719, 1723  
 \MT@set@listname ..... 1190,  
     1352, 1499, 1571, 1618, 1697, 1902, 2789  
 \MT@set@lsbasefont ..... 1971  
 \MT@set@lsfont ..... 1837, 1928, 1940  
 \MT@set@named@keys 3381, 3397, 3419, 3440, 3454, 3463  
 \MT@set@pr@codes ..... 1172  
 \MT@set@pr@heirs ..... 1259, 1448  
 \MT@set@sp@codes ..... 1610  
 \MT@set@sp@heirs ..... 1647, 1651  
 \MT@set@tr@codes ..... 1774, 1935, 1945  
 \MT@set@tr@zero ..... 1785, 1971, 4299  
 \MT@SetExpansion ..... 3393, 3395  
 \MT@SetExtraKerning ..... 3450, 3452  
 \MT@SetExtraSpacing ..... 3436, 3438  
 \MT@SetProtrusion ..... 3377, 3379  
 \MT@SetTracking ..... 3415, 3417  
 \MT@setup@ ..... 773, 777, 4047, 4048, 4745  
 \MT@setup@contexts ..... 2960, 2976  
 \MT@setup@copies ..... 4368, 4384  
 \MT@setup@expansion ..... 4371, 4408  
 \MT@setup@kerning ..... 4375, 4592, 4635  
 \MT@setup@noligatures ..... 4376, 4670  
 \MT@setup@PDF ..... 4367, 4379, 4389  
 \MT@setup@protrusion ..... 4370, 4392  
 \MT@setup@spacing ..... 4374, 4559, 4636  
 \MT@setup@spacing@check ..... 4577, 4748, 4749  
 \MT@setup@tracking ..... 4372, 4544, 4607, 4634  
 \MT@setup@warntracking ..... 4373, 4639  
 \MT@setupfont .. 972, 2889, 4318, 4321, 4325, 4361  
 \MT@setupfont@hook .....  
     ..... 852, 887, 889, 896, 901, 927, 934, 973  
 \MT@shape ..... 1044, 1155, 2435, 2447  
 \MT@shorthandoff ..... 4696, 4727, 4730  
 \MT@show@pdfannot ..... 106  
 \MT@shrink ..... 316, 1514,  
     4419, 4420, 4427, 4428, 4431, 4435, 4492, 4505  
     ..... 1514, 1526, 1570  
 \MT@shrink@ ..... 332  
 \MT@shrink@default ..... 332  
 \MT@size ..... 649, 653, 654, 1044, 1155  
 \MT@size@name ..... 642, 2462  
 \MT@sp@cc@name ..... 1617, 1619, 1671, 3438  
 \MT@sp@context ..... 2972, 3013, 4581  
 \MT@sp@doc@contexts ..... 3013  
 \MT@sp@factor ..... 307  
 \MT@sp@factor@ ..... 1353  
 \MT@sp@inh@name ..... 1645–1647  
 \MT@sp@max ..... 319  
 \MT@sp@min ..... 319  
 \MT@sp@setname ..... 3243  
 \MT@sp@split@val ..... 1623  
 \MT@sp@unit ..... 313  
 \MT@sp@unit@ ..... 1361, 1674  
 \MT@spacing ..... 998, 1607, 4573  
 \MT@spacingfalse ..... 303  
 \MT@spacingtrue ..... 303  
 \MT@split@codes ..... 1227, 1231  
 \MT@split@name ..... 978, 1044  
 \MT@step ..... 316, 1515, 4422,  
     4424, 4442, 4446, 4449, 4506, 4509, 4510, 4514  
 \MT@step@ ..... 1515, 1526, 1570  
 \MT@stretch ..... 316, 1513, 4416,  
     4417, 4420, 4427, 4429, 4434, 4437, 4491, 4505  
 \MT@stretch@ ..... 1513, 1526, 1570  
 \MT@stretch@default ..... 332, 4417  
 \MT@strip@prefix ..... 2732, 2753, 2779  
 \MT@temp ..... 1209–1212, 1661–1665,  
     1732–1735, 2005, 2009, 2012, 2017, 2020,  
     2025, 2084, 2087, 2093, 2097, 2100, 2103,  
     2108, 2116, 2118, 2124, 2198, 2201, 2218,  
     3189, 3214, 4332, 4337, 4342, 4343, 4345, 4348  
 \MT@tempencoding ..... 3818, 3819, 3829, 3835  
 \MT@tempfamily ..... 3830  
 \MT@tempseries ..... 3831  
 \MT@tempshape ..... 3832  
 \MT@tempsize ..... 3786, 3833, 3842, 3844, 3848  
 \MT@test@ast ..... 2033, 2206, 3072, 3077  
 \MT@textls ..... 49, 2159, 2160, 2162  
 \MT@the@pr@code ..... 1216, 1246, 1252, 1834  
 \MT@the@pr@code@tr ..... 1216, 1834  
 \MT@tikz@setup ..... 843, 866, 953  
 \MT@tlist@break ..... 604, 639, 661, 2423, 3932  
 \MT@toks 295, 1317, 1347, 1566, 2520, 2555, 2582,  
     2654, 2661, 2708, 2745, 2760, 2794, 2806, 2814  
 \MT@tr@cc@name .....  
     ..... 1903–1906, 1910, 1924, 1925, 3420, 3427  
 \MT@tr@context ..... 2972, 3013  
 \MT@tr@doc@contexts ..... 3013  
 \MT@tr@factor@ ..... 1912  
 \MT@tr@font@list ..... 1760  
 \MT@tr@ispace ..... 1808, 1809, 1918  
 \MT@tr@ligatures ..... 1824, 1921, 1984, 1987  
 \MT@tr@max ..... 319, 2184–2186  
 \MT@tr@min ..... 319, 2188–2190  
 \MT@tr@noligatures ..... 1824, 1981  
 \MT@tr@okern ..... 1822, 1920  
 \MT@tr@ospace ..... 1811, 1812, 1918  
 \MT@tr@outer@icr ..... 2097, 2126  
 \MT@tr@outer@icr@ ..... 2126  
 \MT@tr@outer@l ..... 1845, 2049  
 \MT@tr@outer@next ..... 2077, 2081, 2128  
 \MT@tr@outer@r ..... 1872, 2077, 2093, 2100, 2126  
 \MT@tr@outer@re ..... 2078, 2083  
 \MT@tr@set@okern ..... 1823, 2194  
 \MT@tr@set@okern@ ..... 2199, 2200, 2205  
 \MT@tr@set@space ..... 1814, 2001  
 \MT@tr@set@space@ ..... 2006–2008, 2014  
 \MT@tr@set@space@@ ..... 2019, 2024, 2027, 2032  
 \MT@tr@setname ..... 3243  
 \MT@tr@unit@ ..... 1905, 2034, 2207  
 \MT@tracking 986, 1760, 1935, 2891, 4299, 4300, 4554  
 \MT@tracking@ ..... 1760, 4300  
 \MT@trackingfalse ..... 305, 4611  
 \MT@trackingtrue ..... 305  
 \MT@try@order ..... 2418, 2426  
 \MT@undefined@char ..... 2701, 2703, 2713  
 \MT@upper ..... 3099, 3105, 3505  
 \MT@use@set ..... 3230, 3235, 3243

- `\MT@UseMicrotypeSet` ..... 3226, 3228  
`\MT@val` ..... 2989–2991, 2993, 2999,  
3001, 3004, 3060, 3064, 3072, 3074, 3094–  
3096, 3108, 3111, 3113, 3116, 3118, 3119,  
3125, 3127, 3128, 3130, 3133, 3135, 3146,  
3149, 3153, 3156–3158, 3160, 3162, 3169–  
3171, 3174, 3183–3185, 3197, 3199, 3204,  
3206, 3208, 3210, 3477, 3479, 3488, 3491,  
3493, 3501–3503, 3513–3515, 3521, 3526,  
3764, 3771, 3775, 3776, 3953, 3954, 3957,  
3965, 3966, 3969, 3971, 3974, 3977, 3996,  
3997, 3999, 4001, 4203, 4204, 4206, 4213,  
4217, 4224, 4230, 4276, 4277, 4279, 4285, 4288  
`\MT@variants` ..... 2411, 3285, 3296  
`\MT@vinfo` ..... 76, 93, 992, 1084, 1356,  
1365, 1369, 1404, 1574, 1591, 1777, 1787,  
2266, 2277, 2334, 2358, 2368, 2371, 2475,  
2477, 2483, 2970, 2993, 3323, 3329, 4086,  
4093, 4209, 4215, 4221, 4228, 4282, 4287, 4689  
`\MT@warn@ascii` ..... 2663, 2793  
`\MT@warn@axis@empty` ..... 3196, 3207, 3212  
`\MT@warn@code@too@large` ..... 1331, 1334, 1338  
`\MT@warn@err` ..... 76, 4090, 4091  
`\MT@warn@ex@too@large` ..... 1550, 1553, 1564  
`\MT@warn@maybe@inputenc` ... 2256, 2807, 2815, 2818  
`\MT@warn@nodim` ..... 3624, 3637, 3647, 3661  
`\MT@warn@number@too@large` ..... 2690, 2799  
`\MT@warn@preset@towitz` ... 1462, 1482, 1675, 1745  
`\MT@warn@rest` ..... 2561, 2598, 2804  
  
**N** `newpx` (package) ..... 21, 142, 234  
`newtx` (package) ..... 21, 142, 234  
  
**O** `\outputmode` ..... 191, 4029  
  
**P** `\pdfadjustinterwordglue` ..... 4301, 4563  
`\pdfadjustspacing` ..... 4260, 4264, 4525  
`\pdfappendkern` ..... 4303, 4596  
`\pdfcopyfont` ..... 1012, 1023  
`\pdfcprot` (package) ..... 26, 30, 54, 63, 97, 226  
`\pdffontexpand` ..... 1524, 1526  
`\pdfnoligatures` ..... 2252, 2268, 2274  
`\pdfprependkern` ..... 4302, 4595  
`\pdfprotrudechars` ..... 192, 4259, 4263, 4658  
`pdfTeX` (engine) ..... 1,  
4, 6–8, 13–19, 23–29, 31, 32, 34, 35, 37–40,  
42, 43, 45, 46, 52, 57, 58, 63, 69, 71, 72,  
74, 75, 78, 79, 82, 84, 110, 113–115, 117,  
126, 130, 132–134, 136, 137, 193, 195, 226–232  
`pdftexcmds` (package) ..... 37, 232  
`\pdftracingfonts` ..... 351  
`\pickup@font` ..... 2861, 2863, 2872, 2895–2897  
`pifont` (package) ..... 98  
`pinyin` (package) ..... 56, 231, 232  
`\protrudechars` ..... 192, 4259  
`protrusion` (option) ..... 6, 11, 123  
`psnfss` (package) ..... 78, 194, 234  
`pstricks` (package) ..... 9, 132  
`pxfonts` (package) ..... 21, 141, 142  
  
**Q** `qfonts` (package) ..... 21, 141, 228  
  
**R** `ragged2e` (package) ..... 26, 80, 135, 232, 233  
`reledmac` (package) ..... 52, 234  
`reysize` (package) ..... 106  
  
**S** `selected` (option) ..... 8, 14, 32, 68, 89, 123, 227  
`\selectfont` ..... 1936, 2964, 2974, 4196, 4740  
`\set@fontsize` ..... 3153  
`\SetExpansion` ..... 14, 36, 3390  
`\SetExtraKerning` ..... 18, 38, 3448  
`\SetExtraSpacing` ..... 19, 39, 3433  
`\SetProtrusion` ..... 13, 35, 3374  
`\SetTracking` ..... 15, 37, 3413  
`\sfcode` ..... 4580  
`shapepar` (package) ..... 29, 53  
`\shbscode` ..... 1642, 1643, 1654, 1657, 1664  
  
`\MT@warn@tracking@DVI` ..... 1788, 4641  
`\MT@warn@unknown` .... 2551, 2567, 2577, 2602, 2812  
`\MT@warn@unknown@once` ..... 2830, 2831  
`\MT@warning` ..... 67, 76, 94, 2185, 2189,  
2269, 2486, 3047, 3160, 3213, 3308, 3331,  
3428, 3557, 3626, 3674, 3688, 3836, 3857,  
3928, 4090, 4095, 4191, 4268, 4307, 4309, 4310  
`\MT@warning@nl` ..... 76, 88, 95, 272, 765,  
769, 805, 817, 903, 1197, 1309, 1345, 1397,  
1437, 1483, 1565, 1993, 2794, 2800, 2805,  
2813, 2862, 3400, 3599, 3939, 3946, 3957,  
4039, 4091, 4096, 4130, 4137, 4172, 4186,  
4358, 4447, 4493, 4512, 4566, 4582, 4647, 4734  
`\MT@while@num` ..... 670, 677, 683, 930–932  
`\MT@with@babel@and@T` .....  
... 782, 855–857, 892–894, 4722–4726, 4729  
`\MT@with@package@T` ... 779, 830, 834, 859, 862–  
864, 866, 871, 878–881, 884, 899, 911, 925,  
926, 928, 942, 953, 954, 2830, 2831, 2873, 4565  
`\MT@xadd` ..... 104, 570, 2359, 2364, 2369,  
2372, 2934, 2943, 3001, 3097, 3325, 3504, 3771  
`\MT@xaddb` ..... 578, 3524, 3846  
`\MT@xdef@n` .. 386, 3245, 3248, 3277, 3282, 3311,  
3427, 3519, 3541, 3701, 3737, 3867, 3954, 3956  
`\MT@xetex@no` ..... 240  
`\MT@xspace` ..... 2116, 2131  
`\MT@xspace@` ..... 2131  
`\MT@xunicodefalse` ..... 829  
`\MT@xunicodetrue` ..... 829, 830, 881  
  
`\normalfont` ..... 971, 4522, 4526  
`\nullfont` ..... 971

<b>T</b>	<code>\tagcode</code> . . . . .	1828, 2255, 2261	<code>tikzposter</code> (package) . . . . .	233
	<code>tempora</code> (package) . . . . .	21, 142, 234	<code>tipa</code> (package) . . . . .	98
	<code>tex4ht</code> (package) . . . . .	54, 230	<code>trace</code> (package) . . . . .	99, 230
	<code>T<sub>E</sub>X Live</code> (distribution) . . . . .	28, 38, 39, 228	<code>\tracingmicrotype</code> . . . . .	87
	<code>\textls</code> . . . . .	23, 52, 53, 919, 944, 949, 1848, 1932, 1948, 2157	<code>\tracingmicrotypeinpdf</code> . . . . .	98
	<code>\textls*</code> . . . . .	23	<code>\tracingmicrotypeinpdfallfalse</code> . . . . .	105
	<code>\textmicrotypecontext</code> . . . . .	22, 48, 915, 2968	<code>\tracingmicrotypeinpdfalltrue</code> . . . . .	105
	<code>tikz</code> (package) . . . . .	53, 222, 232, 234	<code>tracking</code> (option) . . . . .	6, 11, 23, 31, 123
<b>U</b>	<code>ucs</code> (package) . . . . .	94, 231	<code>txfonts</code> (package) . . . . .	21, 141, 142
	<code>ulgothic</code> (package) . . . . .	21	<code>unit</code> (option) . . . . .	7, 32, 228, 230
	<code>\UnicodeEncodingName</code> . . . . .	2772	<code>\UseMicrotypeSet</code> . . . . .	12, 33, 60, 73, 3041, 3224
<b>V</b>	<code>verbose</code> (option) . . . . .	9, 31, 32, 125, 227, 228, 231		
<b>W</b>	<code>wordcount</code> (package) . . . . .	51, 231		
<b>X</b>	<code>XCharter</code> (package) . . . . .	21, 142, 234	<code>\XeTeXglyphindex</code> . . . . .	2575
	<code>xeCJK</code> (package) . . . . .	27, 98, 233, 234	<code>\XeTeXglyphname</code> . . . . .	2594
	<code>X<sub>Y</sub>T<sub>E</sub>X</code> (engine) . . . . .	1, 4, 6–8, 12, 13, 20, 26, 29, 30, 34, 38, 40, 50, 52–54, 82, 92, 130, 134–136, 141, 143, 147, 155, 233, 234	<code>\XeTeXprotrudechars</code> . . . . .	4267
	<code>\XeTeXcharglyph</code> . . . . .	2588	<code>\xspace</code> . . . . .	2115, 2116
	<code>\XeTeXcountglyphs</code> . . . . .	712	<code>xspace</code> (package) . . . . .	82, 233
	<code>\XeTeXfonttype</code> . . . . .	2539	<code>\xspaceskip</code> . . . . .	2063, 2066
	<code>\XeTeXglyph</code> . . . . .	1280	<code>xunicode</code> (package) . . . . .	95, 96, 233
<b>Y</b>	<code>yfonts</code> (package) . . . . .	24	<code>xunicode-addon</code> (package) . . . . .	98
<b>Z</b>	<code>zefonts</code> (package) . . . . .	21, 141		

## 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 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.

### 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

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

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 ‘c<sub>f</sub>gguide.tex’ and ‘m<sub>o</sub>dguide.tex’ in the base L<sup>A</sup>T<sub>E</sub>X distribution for suggestions.

**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 nom-

ination 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 Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
  - (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
  - (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.
  - (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
  - (d) You distribute at least one of the following with the Derived Work:
    - i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
    - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived

Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.

10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
- (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient

documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

## No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any au-

thor named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

## Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
  - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
  - (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current

Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a L<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.