

# The microtype package

An interface to the micro-typographic extensions of pdf $\text{\TeX}$

R Schlicht  
[w.m.l@gmx.net](mailto:w.m.l@gmx.net)

v2.3c  
2008/11/11

---

## Abstract

The `microtype` package provides a  $\text{\LaTeX}$  interface to the micro-typographic extensions of pdf $\text{\TeX}$ : most prominently, character protrusion and font expansion, furthermore the adjustment of interword spacing and additional kerning, as well as hyphenatable letterspacing (tracking) and the possibility to disable all or selected ligatures. It allows to apply these features to customisable sets of fonts, and to configure all micro-typographic aspects of the fonts in a straight-forward and flexible way. Settings for various fonts are provided.<sup>1</sup>

Note that font expansion and character protrusion will only work with pdf $\text{\TeX}$ , at least version 0.14f. Automatic font expansion requires version 1.20 or newer. Disabling ligatures requires pdf $\text{\TeX}$  1.30, letterspacing and the adjustment of interword spacing and of kerning requires version 1.40. The package will by default enable protrusion and expansion if they can safely be assumed to work. These two features are also available with lua $\text{\TeX}$ .

The alternative package `letterspace`, which also works with plain  $\text{\TeX}$ , provides the user commands for letterspacing only, omitting support for all other extensions (see section 7).

This package is copyright © 2004–2008 R Schlicht. It may be distributed and/or modified under the conditions of the  [\$\text{\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’.

---

<sup>1</sup> Currently, this package provides protrusion settings for Computer Modern Roman, Palatino, Times, URW Garamond, Adobe Garamond and Minion, Bitstream Charter and Letter Gothic, the AMS symbols and Euler fonts, for various Euro symbol fonts, as well as some generic settings for unknown fonts (cf. table 3 on page 21). Contributions are very welcome.

## Contents

<b>1</b>	<b>Micro-typography with pdf<math>\TeX</math></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 . . . . .	8
3.4	Tracking/letterspacing . . . . .	9
3.5	Miscellaneous options . . . . .	9
3.6	Changing options later . . . . .	10
<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 . . . . .	16
5.4	Interword spacing . . . . .	18
5.5	Additional kerning . . . . .	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>24</b>
<b>8</b>	<b>Disabling ligatures</b>	<b>25</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>33</b>
14.1	Preliminaries . . . . .	34
	Debugging [35] Requirements [37] Declarations [39] Auxiliary macros [40] Compatibility [48]	
14.2	Font Setup . . . . .	52
	Protrusion [57] Expansion [64] Interword spacing (glue) [66] Additional kerning [68] Tracking [69] Disabling ligatures [77] Loading the configuration [78] Translating characters into slots [82] Hook into $\LaTeX$ 's font selection [87] Context-sensitive setup [90]	

14.3	Configuration . . . . .	92
	Font sets [92] Variants and aliases [98] Disabling ligatures [99] Inter- action with babel [99] Fine tuning [99] Character inheritance [106] Permutation [107]	
14.4	Package options . . . . .	111
	Declaring the options [111] Reading the configuration file [115] Hook for other packages [116] Changing options later [116] Processing the options [119]	
<b>15</b>	<b>Configuration files</b>	<b>126</b>
15.1	Font sets . . . . .	126
15.2	Font variants and aliases . . . . .	127
15.3	Interaction with babel . . . . .	128
15.4	Note on admissible characters . . . . .	129
15.5	Character inheritance . . . . .	129
	OT1 [129] T1 [130] LY1 [130] OT4 [131] QX [131] T5 [132] Euro symbols [133]	
15.6	Tracking . . . . .	133
15.7	Font expansion . . . . .	133
15.8	Character protrusion . . . . .	135
	Default [135] Italics [143] Small caps [151] Italic small caps [154] Text companion [156] Computer Modern math [161] AMS symbols [164] Euler [168] Euro symbols [172]	
15.9	Interword spacing . . . . .	173
	Nonfrenchspacing [175]	
15.10	Additional kerning . . . . .	176
	French [176] Turkish [177]	
<b>16</b>	<b>Auxiliary file for micro fine tuning</b>	<b>177</b>
<b>A</b>	<b>Change history</b>	<b>180</b>
<b>B</b>	<b>Index</b>	<b>187</b>
<b>C</b>	<b>The L<sup>A</sup>T<sub>E</sub>X Project Public License</b>	<b>195</b>

## List of Tables

1	Availability of micro-typographic features . . . . .	7
2	Predefined font sets . . . . .	12
3	Fonts with tailored protrusion settings . . . . .	21
4	Order for matching font attributes . . . . .	80

# 1 Micro-typography with pdf $\text{\TeX}$

pdf $\text{\TeX}$ , the  $\text{\TeX}$  extension written by Hàn Thế Thành, introduces a number of micro-typographic features that make it 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's 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]

Both these features have been lacking a simple  $\text{\LaTeX}$  user interface for quite some time. Then, the `pdfcprot` package was released, which allowed  $\text{\LaTeX}$  users to employ character protrusion without having to mess much with the internals.

Font expansion, however, was still most difficult to utilise, since it required that the font metrics are available for all levels of expansion. Therefore, anybody who wanted to make use of this feature had to create multiple instances of the fonts in advance. Shell scripts to partly relieve the user from this burden were available – however, it remained a cumbersome task. Furthermore, all fonts were still being physically created, thus wasting compilation time and disk space.

In the summer of 2004, Hàn Thế Thành implemented a feature that has proven as a major facilitation for  $\text{\TeX}$  and  $\text{\LaTeX}$  users: font expansion can now take place automatically. That is, pdf $\text{\TeX}$  no longer needs the expanded font metrics but will calculate them at run-time and completely in memory.

After this great leap in usability had been taken, the development did not stop. On the contrary, pdf $\text{\TeX}$  was extended with even more features: version 1.30 introduced the possibility to *disable all ligatures*, version 1.40 a robust *letterspacing* command, the *adjustment of interword spacing* and the possibility to specify *additional character kerning*.

Robust and hyphenatable *letterspacing* (*tracking*) has always been extremely difficult to achieve in  $\text{\TeX}$ . Although the `soul` package undertook great efforts in making this possible, it could still fail in certain circumstances; even to adjust the tracking of a font throughout the document remained impossible. Employing pdf $\text{\TeX}$ 's new extension, this no longer poses a problem. The `microtype` package

provides the possibility to change the tracking of customisable sets of fonts, e. g., all small capitals. It also introduces two new commands `\textls` and `\lststyle` for ad-hoc letterspacing, which can be used like the normal text commands. Note that letterspacing only works in PDF mode.

*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 T<sub>E</sub>X’s ‘space factors’. However, 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 – pdfT<sub>E</sub>X provides the possibility to 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. 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 have an influence on the interword space. Also, the settings that are shipped with microtype are but a first approximation, and I would welcome corrections and improvements very much. I suggest reading the reasoning behind the settings in section 15.9.

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

The possibility, finally, to *disable all ligatures* of a font may be useful for type-writer 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 manner. The next chapters will present a survey of all options and customisation possibilities.

## 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 (which would seem unlikely, 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 respective micro-typographic feature, 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’. Some 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’ll 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 L<sup>A</sup>T<sub>E</sub>X packages, the `microtype` package accepts options in the well known key=value syntax. In the following, you’ll 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 pdf<sub>T</sub><sub>E</sub>X 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 be enabled, font expansion will only be disabled in circumstances where pdf<sub>T</sub><sub>E</sub>X cannot expand the fonts automatically, that is, if it is either too old (versions before 1.20) or if the output mode is DVI (see section 3.5). In other words, `microtype` will try to apply as much micro-typography as can safely be expected to work under the respective conditions (and 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 pdf<sub>T</sub><sub>E</sub>X):

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

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

```
\usepackage{microtype}
```

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	★	★	★	☒	☒	☒ <sup>a</sup>
luaT <sub>E</sub> X	≥ 0.25	DVI	★	☒	⊘	⊘	⊘	⊘
		PDF	★	★	★	⊘	⊘	⊘

★ = enabled    ☒ = not enabled    ⊘ = not available                      <sup>a</sup> ≥ 1.40.4 recommended

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.

**spacing** There is no compatibility level for the new extensions of tracking, interword spacing, and additional kerning. Therefore, they can only be switched on or off, or they may be activated by passing a set name to the option. By default, neither feature is enabled.

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

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,  $\langle dimension \rangle$  character

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

### 3.3 Font expansion

**auto** true, false \* true

As noted in chapter 1, the expanded versions of the fonts may be calculated automatically. This option is true by default provided that pdfTeX's version is found to be 1.20 or higher and the output mode is PDF; otherwise, it will be disabled. If auto is set to false, the fonts 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](#)).

Automatic font expansion does not work with bitmap fonts. Therefore, if you are using the Computer Modern Roman fonts in T1 encoding<sup>2</sup>, you should either install the cm-super fonts or use the Latin Modern fonts (package `lmodern`).

**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**  $\langle integer \rangle$   $\min(\text{stretch}, \text{shrink})/5$

Font expansion will be applied in discrete steps. For example, if step is set to 4 (which it is by default), pdfTeX will try up to eleven different expansion levels of a font (from  $-20$  to  $+20$ ). If you set stretch or shrink to something other than their default values but do not specify step, it will be set to 1/5th of the smaller value of the two. Therefore, the following lines are all equivalent:

```
\usepackage[stretch=20,shrink=20]{microtype}
```

```
\usepackage[stretch=20,step=4]{microtype}
```

```
\usepackage{microtype}
```

**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 to increase 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.

<sup>2</sup> En passant, it may be noted that Type 1 format and T1 encoding are in no other way related than that both start with a 'T' and end with a '1'.



### 3.4 Tracking/letterspacing

**letterspace** *<integer>* 100

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

### 3.5 Miscellaneous options

**DVIoutput** true, false \* false

pdf $\TeX$  is 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.

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 (`dvips`, `dvipdfm`, etc.) resp. the DVI viewer are not able to generate the fonts on the fly.

**draft** true, false false

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

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

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

**babel** true, false false

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

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

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

3 Recent  $\TeX$  systems are using pdf $\TeX$  as the default engine even for DVI output.

### 3.6 Changing options later

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

Inside the preamble, this command accepts all package options described above (except for `config`). In the document body, this command may be used to change the general settings of the micro-typographic extensions. It then accepts all options from section 3.1: `expansion`, `protrusion` and `activate`, which in turn may receive the values `true`, `false`, `compatibility` or `nocompatibility`, and `tracking`, `spacing` and `kerning` 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 that are being 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,LY1,OT4,QX,T5},
  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 OT1, T1, LY1, OT4, QX or T5, 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,LY1,OT4,QX,T5,TS1} }
```

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 expand to the document’s  $\langle value \rangle default$ , e. g.,  $\rm default$ . A single asterisk means  $\langle attribute \rangle default$ , e. g.,  $\encoding default$ , respectively  $\normal size$  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 boundary is included in the range, the upper boundary is not. Thus, ‘12-16’ would match 12 pt, 13.5 pt and 15.999 pt, for example, but not 16 pt. You are allowed to omit the lower or upper bound (‘-10’, ‘large-’).

Additionally to this declaration scheme, you can add single fonts to a set using the ‘font’ key, which expects the concatenation of all font attributes, separated by forward slashes, i. e., ‘font =  $\langle encoding \rangle / \langle family \rangle / \langle series \rangle / \langle shape \rangle / \langle size \rangle$ ’. 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 allowed 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 nine predefined font sets. They may also be activated by passing their name to the feature options protrusion, expansion, tracking, spacing and kerning 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

Table 2: Predefined font sets

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

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, spacing or kerning 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.

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). The only difference is that asterisked values will be expanded 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 both settings for the current family (say, T1/cmr//) and settings for italic fonts in the normal weight (T1//m/it/) exist, those for the Computer Modern Roman font would apply.<sup>4</sup> The encoding must always match.

<sup>4</sup> For the interested, table 4 on page 80 presents the exact order.

## 5.1 Character protrusion

`\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 characters may be specified either as a single character (‘A’), as a text symbol command (‘\textquoteleft’), or as a slot number: three digits for decimal notation, prefixed with ‘#’ for hexadecimal, with ‘o’ for octal (e. g., the ‘fl’ ligature in T1 encoding: 029, #1D, o35). 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. Note that you also have the possibility to declare lists of characters that should inherit settings (see section 5.6).

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 can 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 previously assigned a name to it) before the current list will be loaded, so that the fonts will inherit the values from the loaded list.

Thus, 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:

**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 an example application, see section 6.

## 5.2 Font expansion

`\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 was 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 package was loaded with the selected option, 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 that all characters of a particular font (*set*) 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.

<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.

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 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 easily be avoided by shrinking the font a little 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 an 'unnecessary' widow.}
```

This method of employing contexts to temporarily apply different expansion parameters only works with pdfTeX version 1.40.4 or later (for older versions, a dirty trick is laid out in section 14.2 on page 54). 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

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

An important typographic technique – which was missing in T<sub>E</sub>X for a long time – is the adjustment of tracking, i. e., the uniform addition or subtraction of letter space 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>6</sup> The `\SetTracking` command allows to specify 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 0.166 em, 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

<sup>6</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.



default equal to the current letterspace amount. To remove kerning on both sides, you would write ‘outer kerning={0,0}’.

**no ligatures** As far as pdf $\TeX$  is concerned, ligatures in letterspaced fonts would 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. This is not recommended, however, since it also entails that kerning will be switched off.<sup>7</sup> The default settings disable ligatures for the character ‘f’ only, i.e., ‘ff’, ‘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 \lslig (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 I would in no way recommend; they are only for 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 is the (typographically dubious) outcome:

Stop stealing sheep!

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

<sup>7</sup> The inseparable connexion of ligatures and kerns is a limitation of  $\TeX$  that will not be lifted before the advent of lua $\TeX$ .

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

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

As another, more realistic example, suppose you want to space out all small capitals by 50/1000 em, fonts smaller than `\small` by 0.02 em, and to decrease the tracking of large type by 0.02 em. You can achieve this 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 0.1 em (adjustable with the package option `letterspace`, see section 3.5). Suppose 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 Interword spacing

`\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,  $\text{\TeX}$  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  $\text{\TeX}$  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,  $\text{\TeX}$  has the concept of 'space factors'. They may be used to increase the space after certain characters, most prominently the punctuation characters. If pdf $\text{\TeX}$ 's additional spacing adjustment is in effect, space factors are ignored, since it may be considered an extension to space factors with much finer control.

The *spacing settings* are declared as pairs of  $\langle \text{character} \rangle = \langle \text{spacing factors} \rangle$ , 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, however, the settings must 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 these (nonsensical) settings:

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

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

## 5.5 Additional kerning

**\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.

I should not neglect to mention a limitation of this additional kerning: 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 'apos\ -trophe’. This restriction of pdfTeX will hopefully be lifted soon.

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 to apply 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}
```

## 5.6 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 Å, 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, with the only exception that exactly one encoding must be specified. 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.

In the main configuration file `microtype.cfg` and the other font-specific configuration files, you can find examples of all these commands.

## 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

Table 3: Fonts with tailored protrusion settings

Font family (NFSS code)	Features	
	Encodings	Shapes
Generic	OT1, T1, LY1, QX, (TS1) <sup>a</sup>	n, (it, sl, sc) <sup>a</sup>
Computer Modern Roman (cmr) <sup>b</sup>	OT1, OT4, T1, 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) <sup>g</sup>	OT1, T1, LY1, TS1	n, it, (sl) <sup>d</sup> , sc, si
Palatino (ppl, pplx, pplj) <sup>h</sup>	OT1, OT4, T1, LY1, (TS1) <sup>a</sup>	n, it, (sl) <sup>d</sup> , sc
Times (ptm, ptmx, ptmj) <sup>i</sup>	OT1, OT4, T1, LY1, QX, (TS1) <sup>a</sup>	n, it, (sl) <sup>d</sup> , sc
Computer Modern math (cmsy, cmm)	OML/OMS	n/it
AMS symbols (msa, msb)	U	n
Euler (eur, eus, euf) <sup>j</sup>	U	n
Euro symbols (Adobe, ITC, marvosym)	U/OT1	n, it

<sup>a</sup> Incomplete  
<sup>b</sup> Aliases: Latin Modern (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor)  
<sup>c</sup> Aliases: mathdesign/Charter (mdbch), MicroPress's chmath (chr)  
<sup>d</sup> Settings inherited from italic shape  
<sup>e</sup> Alias: mathdesign/URW Garamond (mdugm)  
<sup>f</sup> Alias: ulgothic (ulg)  
<sup>g</sup> By courtesy of Harald Harders ([h.harders@tu-bs.de](mailto:h.harders@tu-bs.de))  
<sup>h</sup> Aliases: pxfonts (pxr), qfonts/QuasiPalatino, T<sub>E</sub>X Gyre Pagella (qpl), FPL Neu (fp9x, fp9j)  
<sup>i</sup> Aliases: txfonts (txr), qfonts/QuasiTimes, T<sub>E</sub>X Gyre Termes (qtm)  
<sup>j</sup> Alias: eulervm (zeur, zeus)

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 are embarking 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`’), 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](#)). This allows it to put settings for, e. g., the 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` {*<font name>*}

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

## 6 Context-sensitive setup

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

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

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

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

`\end{microtypecontext}` Like many  $\LaTeX$  commands, it is also available in the form of an environment.

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

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

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

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

<sup>9</sup> Font package authors might also want to have a look at the hook `\Microtype@Hook`, described in the implementation part, section 14.4.3.

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
```

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

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

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

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

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

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

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

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

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

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

## 7 Letterspacing revisited

`\textls` [*amount*] {*general text*}

`\textls*` 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 in the same way as L<sup>A</sup>T<sub>E</sub>X's text commands (independently of whether the tracking option is enabled): `\textls` expects the text in the mandatory argument, while `\lsstyle` will switch on letterspacing for all subsequent fonts until the end of the current group. 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 em = 0.1 em; this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option, with decreasing significance in this order.

`\lslig` {*ligature*}

Since the commands `\textls` and `\lsstyle` will also evaluate the 'no ligatures' key for the respective font, you need not worry about protecting or breaking ligatures with most fonts. However, in certain situations, there may be a conflict of ligatures beginning with the same letter, where some of them should be inhibited, while others should not. When letterspacing text typeset in Fraktur fonts, for example, the ligatures 'ch', 'ck', 'tz' and 'sz' ('ß') should never be broken up; you also usually see the 'st' ('ſt') ligature in letterspaced text. Furthermore, at least the `yfonts` package realises the short s ('ſ') as the ligature 's:'. On the other hand, the 'ct' ligature and the other 'long s' ligatures often found in Fraktur fonts should be suppressed. There are two ways to solve 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, 'A<sup>u</sup>sſi<sup>ch</sup>t<sup>s</sup>l<sup>o</sup>ſi<sup>g</sup>ſe<sup>i</sup>t').

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

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

`letterspace.sty` These three commands (plus the `letterspace` option, described in section 3.4) are also available with the alternative `letterspace` package, which is in fact a much stripped-down version of `microtype`, omitting support for all the other extensions (and also omitting the possibilities of the `\SetTracking` command – all 'f' ligatures will be disabled, inner and outer spacing and outer kerning will be set to the default values described in section 5.3). If you prefer to forgo `microtype`'s specialties, you

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



may load the `letterspace` package instead. Both packages should not be used at the same time.

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

## 8 Disabling ligatures

`\DisableLigatures` [*characters*] [*set of fonts*]

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 '? !', but not fi – » etc.
```

The character that begins the ligature(s) is what matters. This command may only be used in the preamble, and only once. It requires pdf $\text{\TeX}$  1.30 or newer.

## 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 or expansion. 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.

*Settings for Cyrillic/Greek/Thai 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., T2A, LGR 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.

*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. 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.

*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.

*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 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 choosing a different font set). While the `\microtypesetup` command has of course been designed for cases like this, you might find it tiring to repeat it every time if you are using the `verbatim` environment frequently. The following line, added to the document's preamble, would serve the same purpose:

```
\g@addto@macro\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.

*Compatibility and interaction with other packages:* The `microtype` package is supposed to work happily together with all other L<sup>A</sup>T<sub>E</sub>X packages (except for `pdfcpot`). However, life isn't perfect, so problems are to be expected. Currently, I am aware of the following issues:

- If you want to use 8-bit characters in the configuration, you have to load the `inputenc` package first. Unicode input is also supported (when loading `inputenc` with the `utf8` or the `utf8x` option). 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`.
- 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 the CJK package is font expansion.

*Possible error messages and how to get rid of them:*

- ! 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 pdfT<sub>E</sub>X in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your T<sub>E</sub>X system.
- ! pdfTeX error (font expansion): auto expansion is only possible with scalable fonts.  
Automatic font expansion has been improved in pdfT<sub>E</sub>X 1.40, in that it now not only works with Type 1 fonts but also with TrueType, OpenType and even non-embedded fonts. The above error message indicates either that you are trying to apply expansion to a bitmap (pk) font, which is still not possible, or that the font isn't found at all, e. g., because of missing map entries.
- Warning: 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 pdfT<sub>E</sub>X versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your T<sub>E</sub>X system is not set up to embed (or 'download') the base PostScript fonts (e. g., Times, Helvetica, Courier). In most T<sub>E</sub>X distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to `true`.

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

## 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 pdf<sub>T</sub><sub>E</sub>X programme in the first place, which introduced the micro-typographic extensions and made them available to the <sub>T</sub><sub>E</sub>X world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004](#) and [Thành 2008](#) and elsewhere. I also thank him and the rest of the pdf<sub>T</sub><sub>E</sub>X team for refuting the idea that <sub>T</sub><sub>E</sub>X 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 Bühmann* 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. My thanks also go to *Maciej Eder* for contributing settings for the QX encoding.

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.

Additionally, the following people have reported bugs, made suggestions or helped otherwise (in chronological order): *Tom Kink*, *Herb Schulz*, *Michael Hoppe*, *Gary L. Gray*, *Georg Verweyen*, *Christoph Bier*, *Peter Muthesius*, *Bernard Gaulle* †, *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*, *Geoff Vallis*, *Steven E. Harris*, *Karl Berry*, *Peter Meier*, *Nathan Rosenblum*, *Wolfram Schaalo*, *Vasile Gaburici* and *Sveinung Heggen*.

## 12 References

Hàn Thê Thành, ‘Micro-typographic extensions to the T<sub>E</sub>X typesetting system’, Diss. Masaryk University Brno 2000, in: *TUGBoat*, vol. 21(2000), no. 4, pp. 317–434. (Online at <http://www.tug.org/TUGboat/Articles/tb21-4/tb69thanh.pdf>)

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

Hàn Thê Thành, ‘Font-specific issues in pdfT<sub>E</sub>X’, in: *TUGBoat*, vol. 29(2008), no. 1: ‘EuroBachOT<sub>E</sub>X 2007 Proceedings’, pp. 36–41. (Online – for TUG members – at <https://www.tug.org/members/TUGboat/tb29-1/tb91thanh-fonts.pdf>)

Hàn Thê Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, Martin Schröder, *The pdfT<sub>E</sub>X user manual*, 25 January 2007. (Available from CTAN at [/systems/pdftex/](#); latest version at <http://sarovar.org/projects/pdftex/>)

Karl Berry, *Fontname. Filenames for T<sub>E</sub>X fonts*, September 2005. (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](#))

Carsten Schurig, Tobias Schlemmer, *The pdfcp<sub>rot</sub>.sty package*, 10 June 2005. (Available from CTAN at [/macros/latex/contrib/pdfcp<sub>rot</sub>/](#))

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

## 13 Short history

The comprehensive list of changes can be found in appendix A. The following is a list of all changes relevant in the user land; bug and compatibility fixes are swept under the rug.

### 2.3c (2008/11/11)

- Support for lua $\TeX$  enabled by default

### 2.3 (2007/12/23)

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

### 2.2 (2007/07/14)

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

### 2.1 (2007/01/21)

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

### 2.0 (2007/01/14)

- Support for the new extensions of pdf $\TeX$  version 1.40: tracking/letterspacing, adjustment of interword spacing (glue), and additional kerning (new commands `\SetTracking`, `\SetExtraSpacing`, `\SetExtraKerning`; new options ‘tracking’, ‘spacing’, ‘kerning’) [sections 5.3, 5.4, 5.5]
- New commands `\textls` and `\lststyle` for letterspacing, new option ‘letterspace’ [sections 3.4, 7]
- New option ‘babel’ for automatic micro-typographic adjustment to the selected language [sections 3.5, 6]
- New font sets: ‘smallcaps’, ‘footnotesize’, ‘scriptsize’ [section 4; table 2]
- New package ‘letterspace’ providing the commands for robust and hyphenatable letterspacing [section 7]

## 1.9e (2006/07/28)

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

## 1.9d (2006/05/05)

- Support for the Central European QX encoding (inheritance, generic protrusion settings, contributed by Maciej Eder; protrusion settings for Times)
- 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; consequently, no need to change font defaults before loading microtype, or to put it the other way round, microtype may now be loaded at any time
- Inside the preamble, `\microtypesetup` accepts all package options [section 3.6]
- Protrusion settings for T5 encoded Charter

## 1.9 (2005/10/28)

- New command `\DisableLigatures` to disable ligatures of fonts (requires pdfTeX version 1.30 or later) [section 8]
- New command `\microtypecontext` to change the configuration context; new key ‘context’ for the configuration commands [section 6]
- New key ‘font’ to add single fonts to the font sets [section 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 [section 4]
- New option ‘config’ to load a different configuration file [section 3.5]
- New option ‘unit’ to measure protrusion factors relative to a dimension instead of the character width [section 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
- When using the `ledmac` package, character protrusion will work for the first time ever (requires pdfTeX version 1.30 or later)

## 1.7 (2005/03/23)

- Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings [sections 4, 5]
- New command `\LoadMicrotypeFile` to load a font configuration file manually [section 5.7]

- Hook `\Microtype@Hook` for font package authors [section 14.4.3]
- 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 [sections 3.2, 5]
- When pdf $\TeX$  is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled [section 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 [section 3.1]
- New option `'selected'` to enable selected expansion, default: false [sections 3.3, 5.2]
- New default for expansion option `'step'`: 4 ( $\min(\text{stretch}, \text{shrink})/5$ ) [section 3.3]
- Protrusion settings for Bitstream Charter

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

- Set up fonts independently from  $\LaTeX$  font loading
- New option: `'final'` [section 3.5]

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

- New font sets: `'allmath'` and `'basicmath'` [section 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, contributed by Harald Harders
- New command: `\DeclareCharacterInheritance` [section 5.6]
- Characters may also be specified as octal or hexadecimal numbers [section 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`).

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

`lua`: Code for `luaTeX` (microtype only).

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

`debug`: Code for additional output in the log file.

Used for – surprise! – debugging purposes.

`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 censeo: it should be forbidden for packages to change catcodes within the preamble.

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

```
5 \let\MT@restore@catcodes\@empty
6 \def\MT@fix@catcode#1#2{%
7   \edef\MT@restore@catcodes{%
8     \MT@restore@catcodes
9     \catcode#1 \the\catcode#1\relax
10  }%
11  \catcode#1 #2\relax
12 }
13 <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 pdf $\TeX$ .

```
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*\lststyle{}
52 \newcommand\textls[2][]{\textls[2]{}}
53 \def\textls#1#{}
54 \newcommand*\lslig[1]{#1}
55 *package
56 }

```

These commands also have a starred version.

```

57 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
58 \def\DeclareMicrotypeVariants#1#{\@gobble}

```

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

```

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

```

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

```

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

```

`\MT@warning`      Communicate.

```

\MT@warning@nl 75 \def\MT@warning{\PackageWarning\MT@MT}
\MT@info       76 \def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
\MT@info@nl    77 *package
\MT@vinfo      78 \def\MT@info{\PackageInfo\MT@MT}
\MT@vinfo      79 \def\MT@info@nl#1{\MT@info{#1\@gobble}}
\MT@error      80 \let\MT@vinfo@gobble
\MT@warn@err   81 \def\MT@error{\PackageError\MT@MT}
\MT@warn@err   82 \def\MT@warn@err#1{\MT@error{#1}{%
83   This error message appears because you loaded the \MT@MT'\MessageBreak
84   package with the option `verbose=errors'. Consult the documentation\MessageBreak
85   in \MT@MT.pdf to find out what went wrong.}}

```

### 14.1.1 Debugging

`\tracingmicrotype`      Cases for `\tracingmicrotype`:

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

```

86 <debug>
87 \MT@warning@n1{This is the debug version}
88 \newcount\tracingmicrotype
89 \tracingmicrotype=2
90 \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
91 \def\MT@info@n1#1{\PackageInfo\MT@MT{#1\@gobble}\MT@addto@annot{#1}}
92 \let\MT@vinfo\MT@info@n1
93 \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
94 \def\MT@warning@n1#1{\PackageWarning\MT@MT{#1\@gobble}\MT@addto@annot{Warning: #1}}
95 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
96 \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

```

97 \newcount\tracingmicrotypeinpdf

```

Let's see how it works . . .

```
\tracingmicrotypeinpdf=2
```

\MT@pdf@annot

During font setup, we save the text for the popup in \MT@pdf@annot. (This requires  $\text{pdf}_{\text{TeX}} \geq 1.30$ .)

\MT@addto@annot

\ifMT@inannot

```

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

```

\iftracingmicrotypeinpdfall

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

```

103 \newif\iftracingmicrotypeinpdfall

```

\MT@show@pdf@annot

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).

```

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

```

122 `</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

```

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

```

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

```

142 \def\MT@requires@latex#1{%
143 \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
144 }
145 </plain>

```

`\MT@pdftex@no` pdf<sub>T</sub>E<sub>X</sub>'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 pdf<sub>T</sub>E<sub>X</sub> 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 pdf<sub>T</sub>E<sub>X</sub>:

0: not running pdf<sub>T</sub>E<sub>X</sub>

1: pdf<sub>T</sub>E<sub>X</sub> (< 0.14f)

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

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

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

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

6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`<sup>11</sup>; `\pdftracingfonts`; always e-<sub>T</sub>E<sub>X</sub> (≥ 1.40)

---

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

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

```
146 \def\MT@pdftex@no{0}
```

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

```
147 \ifx\normalpdftexversion\@undefined \else
148   \let\pdftexversion\normalpdftexversion
149   \let\pdftexrevision\normalpdftexrevision
150   \let\pdfoutput\normalpdfoutput
151 \fi
```

Old packages might have let \pdftexversion to \relax.

```
152 \ifx\pdftexversion\@undefined \else
153   \ifx\pdftexversion\relax \else
154     <debug>\MT@info@nl{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
155     \def\MT@pdftex@no{7}
156     <package>
157     \ifnum\pdftexversion = 140
158       \ifnum\pdftexrevision < 4
159         \def\MT@pdftex@no{6}
160       \fi
161     \else
162     </package>
163     \ifnum\pdftexversion < 140
164       \def\MT@pdftex@no{5}
165     <package>
166     \ifnum\pdftexversion < 130
167       \def\MT@pdftex@no{4}
168     \ifnum\pdftexversion < 120
169       \def\MT@pdftex@no{3}
170     \ifnum\pdftexversion = 14
171       \ifnum\expandafter`\pdftexrevision < `h
172         \def\MT@pdftex@no{2}
173       \ifnum\expandafter`\pdftexrevision < `f
174         \def\MT@pdftex@no{1}
175       \fi
176     \fi
177   \else
178     \ifnum\pdftexversion < 14
179       \def\MT@pdftex@no{1}
180     \fi
181   \fi
182 \fi
183 \fi
184 \fi
185 </package>
186 \fi
187 \fi
188 \fi
189 <debug>\MT@info@nl{0}{pdftex no.: \MT@pdftex@no}
```

\MT@clear@options     If we are not using pdfT<sub>E</sub>X or in case it is too old, we disable everything and exit.

```
190 \def\MT@clear@options{%
191   <plain> \MT@requires@latex1{%
192     \AtEndOfPackage{\let\unprocessedoptions\relax}%
193     \let\CurrentOption\empty
194   <plain> } \relax
195 }
196 \ifnum\MT@pdftex@no <
```

```

197 <package> 2
198 <letterspace> 6
199 \MT@warning@nl{You
200 \ifcase\MT@pdfTeX@no
201 don't seem to be using pdfTeX.\MessageBreak
202 ~\MT@MT' only works with pdfTeX.\MessageBreak
203 Try running 'pdflatex' instead of 'latex'%
204 \else
205 are using a pdfTeX version older than
206 <package> 0.14%
207 <letterspace> 1.40%
208 .\MessageBreak
209 ~\MT@MT' does not work with this version.\MessageBreak
210 Please install a newer version of pdfTeX%
211 \fi
212 }
213 \MT@clear@options\MT@restore@catcodes
214 \endinput\fi

```

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

```

215 \RequirePackage{keyval}[1997/11/10]
216 <*package>

```

```

\MT@toks    We need a token register.
217 \newtoks\MT@toks
\ifMT@if@    A scratch if.
218 \newif\ifMT@if@

```

### 14.1.3 Declarations

```

\ifMT@protrusion    These are the global switches ...
\ifMT@expansion 219 \newif\ifMT@protrusion
\ifMT@auto 220 \newif\ifMT@expansion
221 \newif\ifMT@auto
\ifMT@selected 222 \newif\ifMT@selected
\ifMT@noligatures 223 \newif\ifMT@noligatures
\ifMT@draft 224 \newif\ifMT@draft
\ifMT@spacing 225 \newif\ifMT@spacing
\ifMT@kerning 226 \newif\ifMT@kerning
\ifMT@tracking 227 \newif\ifMT@tracking
\ifMT@tracking 228 \newif\ifMT@babel
\MT@MT@babel ... and numbers.
\MT@ex@level 229 \let\MT@pr@level\tw@
\MT@pr@factor 230 \let\MT@ex@level\tw@
231 \let\MT@pr@factor\@m
\MT@ex@factor 232 \let\MT@ex@factor\@m
\MT@sp@factor 233 \let\MT@sp@factor\@m
\MT@kn@factor 234 \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 235 \let\MT@pr@unit\@empty
236 \let\MT@sp@unit\@mone
237 \def\MT@kn@unit{1em}
\MT@stretch    Expansion settings.
\MT@shrink
\MT@step 238 \let\MT@stretch\@mone

```

```

239 \let\MT@shrink \m@ne
240 \let\MT@step \m@ne

\MT@pr@min      Minimum and maximum values allowed by pdfTeX.
\MT@pr@max 241 \def\MT@pr@min{-\@m}
\MT@ex@min 242 \let\MT@pr@max\@m
243 \let\MT@ex@min\z@
\MT@ex@max 244 \let\MT@ex@max\@m
\MT@sp@min 245 \def\MT@sp@min{-\@m}
\MT@sp@max 246 \let\MT@sp@max\@m
247 \def\MT@kn@min{-\@m}
\MT@kn@min 248 \let\MT@kn@max\@m
\MT@kn@max 249 /package
250 \def\MT@tr@min{-\@m}
\MT@tr@min 251 \let\MT@tr@max\@m
\MT@tr@max 252 *package

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

\MT@stretch@default Default values for expansion.
\MT@shrink@default 254 \def\MT@stretch@default{20 }
\MT@step@default 255 \def\MT@shrink@default{20 }
256 \def\MT@step@default{4 }

\MT@letterspace Default value for letterspacing (in thousandths of 1 em).
\MT@letterspace@default 257 /package
258 \let\MT@letterspace\m@ne
259 \def\MT@letterspace@default{100}
260 *package

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

```

#### 14.1.4 Auxiliary macros

```

\MT@maybe@etex For definitions that depend on e-TeX features.
262 \ifcase 0%
263   \ifx\TeXversion\@undefined 1\else
264     \ifx\TeXversion\relax 1\else
265       \ifcase\TeXversion 1\fi
266     \fi
267   \fi
268 \else
269   \catcode\^Q=9 \catcode\^X=14
270 \fi
271 debug\MT@info@nl{0}{this is
272 debug\^Q not
273 debug etex}

\MT@requires@pdftex For definitions that depend on a particular pdfTeX version.
274 \def\MT@requires@pdftex#1{%
275   \ifnum\MT@pdftex@no<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
276 }
277 debug\MT@requires@pdftex6{\pdftracingfonts=1}\relax

\MT@requires@luatex For definitions that depend on luaTeX.
278 *lua
279 \def\MT@requires@luatex{%
280   \ifx\directlua\@undefined\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi

```



```

281 }
282 <debug>\MT@info@n10{this is \MT@requires@luatex{}{not }luatex}

\MT@lua      Communicate with lua.
283 \def\MT@lua{\directlua0}
284 </lua>
285 </package>

    Since luaTeX is included in TeX Live 2008, we now support it by default, even
    though luaTeX still ignores \letterspacefont.

286 <!*lua|letterspace>
287 \ifx\directlua\undefined\else
288 <!letterspace> \MT@error
289 <letterspace> \MT@warning@n1
290   {\MT@MT'
291 <!letterspace>   only works with luatex if you generate%
292 <letterspace>   doesn't currently work with luatex.%
293   \MessageBreak
294 <!letterspace>   the package with the `lua' option%
295 <letterspace>   Bye bye%
296   }
297 <!letterspace> {}
298 <letterspace> \MT@clear@options\MT@restore@catcodes\expandafter\endinput
299 \fi
300 </!*lua|letterspace>

\MT@glet      The forgotten primitive.
301 \def\MT@glet{\global\let}

\MT@exp@cs    Commands to create command sequences. Those that are going to be defined
\MT@exp@gcs    globally should be created inside a group so that the save stack won't explode.
302 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
303 <*package>
304 \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    305 \def\MT@def@n{\MT@exp@cs\def}
306 \def\MT@gdef@n{\MT@exp@gcs\gdef}

\MT@edef@n    Its expanding versions.
\MT@xdef@n    307 </package>
308 \def\MT@edef@n{\MT@exp@cs\edef}
309 <*package>
310 \def\MT@xdef@n{\MT@exp@gcs\xdef}

\MT@let@nc    \let a \csname sequence to a command.
\MT@glet@nc    311 \def\MT@let@nc{\MT@exp@cs\let}
312 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}

\MT@let@cn    \let a command to a \csname sequence.
313 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}

\MT@let@nn    \let a \csname sequence to a \csname sequence.
\MT@glet@nn    314 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}
315 \def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}

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

\MT@exp@one@n Expand the second token once and enclose it in braces.
317 </package>

```

```

318 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}
\MT@exp@two@c      Expand the next two tokens after <#1> once.
319 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
320 <package>
\MT@exp@two@n      Expand the next two tokens after <#1> once and enclose them in braces.
321 \def\MT@exp@two@n#1#2#3{%
322   \expandafter\expandafter\expandafter
323   #1\expandafter\expandafter\expandafter
324   {\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 325 \def\MT@ifdefined@c@T#1{%
326   ^^X \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
327   ^^Q \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
328 }
329 </package>
330 \def\MT@ifdefined@c@TF#1{%
331   ^^X \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
332   <package>^^Q \ifx#1\@undefined
333   <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
334 }
335 \def\MT@ifdefined@n@T#1{%
336   ^^X \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
337   <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
338   <package>^^Q \expandafter\@gobble\else\expandafter\@firstofone\fi
339 }
340 <package>
341 \def\MT@ifdefined@n@TF#1{%
342   ^^X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
343   ^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
344   ^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
345 }
\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).
346 \def\MT@detokenize@n#1{%
347   ^^X \expandafter\MT@rem@last@space\detokenize{#1} \@nil
348   ^^Q \string#1%
349 }
350 \def\MT@detokenize@c#1{%
351   ^^X \MT@exp@one@n\MT@detokenize@n#1%
352   ^^Q \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil
353 }
354 \def\MT@rem@last@space#1 #2{#1%
355   \ifx\@nil#2\else \space
356   \expandafter\MT@rem@last@space\expandafter#2\fi
357 }
\MT@ifempty      Test whether argument is empty.
358 <package>
359 \begingroup
360 \catcode`\%=12
361 \catcode`\&=14
362 \gdef\MT@ifempty#1{&
363   \if %#1%&

```

```

364     \expandafter\@firstoftwo
365     \else
366     \expandafter\@secondoftwo
367     \fi
368 }
369 \endgroup
370 <*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).

371 \MT@requires@pdftex6{
372   <*lua>
373   \MT@requires@luatex{
374     \def\MT@ifint#1{%
375       \csname \MT@lua{
376         if string.find("\luaescapestring{#1}", "^-[0-9]+ *$")
377         then tex.write("@firstoftwo")
378         else tex.write("@secondoftwo")
379         end}%
380       \endcsname
381     }
382   }{
383     </lua>
384     </package>
385     \def\MT@ifint#1{%
386       \ifcase\pdfmatch{^-[0-9]+ *$}{#1}\relax
387       \expandafter\@secondoftwo
388       \else
389       \expandafter\@firstoftwo
390       \fi
391     }
392     <*package>
393     <lua> }
394   }{
395     \def\MT@ifint#1{%
396       \if!\ifnum9<1#1!\else?\fi
397       \expandafter\@firstoftwo
398       \else
399       \expandafter\@secondoftwo
400       \fi
401     }
402   }

\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.)

403 \MT@requires@pdftex6{
404   <*lua>
405   \MT@requires@luatex{
406     \def\MT@ifdimen#1{%
407       \csname \MT@lua{
408         if (string.find("\luaescapestring{#1}", "^-[0-9]+(\@percentchar a*) *$") or
409         string.find("\luaescapestring{#1}", "^-[0-9]*[.][0-9]+(\@percentchar a*) *$"))
410         then tex.write("@firstoftwo")
411         else tex.write("@secondoftwo")
412         end}%
413       \endcsname
414     }
415   }{
416     </lua>
417     \def\MT@ifdimen#1{%
418       \ifcase\pdfmatch{^([0-9]+([.][0-9]+)?|.[.][0-9]+)%

```

```

419             (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
420         \expandafter\@secondoftwo
421     \else
422         \expandafter\@firstoftwo
423     \fi
424 }
425 {lua} }
426 }{
427     \def\MT@ifdimen#1{%
428         \setbox\z@=\hbox{%
429             \MT@count=1#1\relax
430             \ifnum\MT@count=\@ne
431                 \aftergroup\@secondoftwo
432             \else
433                 \aftergroup\@firstoftwo
434             \fi
435         }%
436     }
437 }

\MT@ifdim    Test floating point numbers.

438 \def\MT@ifdim#1#2#3{%
439     \ifdim #1\p@ #2 #3\p@
440         \expandafter\@firstoftwo
441     \else
442         \expandafter\@secondoftwo
443     \fi
444 }

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

445 \MT@requires@pdftex5{
446 {*lua}
447     \MT@requires@luatex{
448         \def\MT@ifstreq#1#2{%
449             \csname \MT@lua{
450                 if "\luaescapestring{#1}" == "\luaescapestring{#2}"
451                     then tex.write("@firstoftwo")
452                     else tex.write("@secondoftwo")
453                 end}%
454             \endcsname
455         }
456     }{
457 {/lua}
458         \def\MT@ifstreq#1#2{%
459             \ifcase\pdfstrcmp{#1}{#2}\relax
460                 \expandafter\@firstoftwo
461             \else
462                 \expandafter\@secondoftwo
463             \fi
464         }
465 {lua} }
466 }{
467     \def\MT@ifstreq#1#2{%
468         \edef\MT@res@a{#1}%
469         \edef\MT@res@b{#2}%
470         \ifx\MT@res@a\MT@res@b
471             \expandafter\@firstoftwo
472         \else
473             \expandafter\@secondoftwo
474         \fi
475     }
476 }

```

`\MT@xadd`      Add item to a list.

```
477 \def\MT@xadd#1#2{%
478   \ifx#1\relax
479     \xdef#1{#2}%
480   \else
481     \xdef#1{#1#2}%
482   \fi
483 }
```

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

```
484 \def\MT@xaddb#1#2{%
485   \ifx#1\relax
486     \xdef#1{#2}%
487   \else
488     \xdef#1{#2#1}%
489   \fi
490 }
```

`\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@` 491 *</package>*

`\MT@clist@function` 492 `\def\MT@map@clist@n#1#2{%`

`\MT@clist@break` 493 `\ifx\@empty#1\else`

494 `\def\MT@clist@function##1{#2}%`

495 `\MT@map@clist@#1,\@nil,\@nnil`

496 `\fi`

497 `}`

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

499 `\def\MT@map@clist@#1,{%`

500 `\ifx\@nil#1%`

501 `\expandafter\MT@clist@break`

502 `\fi`

503 `\MT@clist@function{#1}%`

504 `\MT@map@clist@`

505 `}`

506 `\let\MT@clist@function\@gobble`

507 `\def\MT@clist@break#1\@nnil{}`

508 *<\*/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@` 509 `\def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}`

`\MT@tlist@break` 510 `\def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}`

511 `\def\MT@map@tlist@#1#2{%`

512 `\ifx\@nnil#2\else`

513 `#1{#2}%`

514 `\expandafter\MT@map@tlist@`

515 `\expandafter#1%`

516 `\fi`

517 `}`

518 `\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`

519 `\newif\ifMT@inlist@`

520 `\def\MT@in@clist#1#2{%`

521 `\def\MT@res@a#1,#1,##2##3\@nnil{%`

522 `\ifx##2\@empty`

523 `\MT@inlist@false`

524 `\else`

525 `\MT@inlist@true`

526 `\fi`

```

527 }%
528 \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
529 }

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

530 \def\MT@rem@from@clist#1#2{%
531   \def\MT@res@a##1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
532   \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
533   \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
534 }

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

535 \def\MT@in@tlist#1#2{%
536   \MT@inlist@false
537   \def\MT@res@a{#1}%
538   \MT@map@tlist@c#2\MT@in@tlist@
539 }
540 \def\MT@in@tlist@#1{%
541   \edef\MT@res@b{#1}%
542   \ifx\MT@res@a\MT@res@b
543     \MT@inlist@true
544     \expandafter\MT@tlist@break
545   \fi
546 }

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

\MT@in@rlist@@ 547 \def\MT@in@rlist#1{%
\MT@size@name 548   \MT@inlist@false
549   \MT@map@tlist@c#1\MT@in@rlist@
550 }
551 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
552 \def\MT@in@rlist@@#1#2#3{%
553   \MT@ifdim{#2}=\m@ne{%
554     \MT@ifdim{#1}=\MT@size
555     \MT@inlist@true
556     \relax
557   }{%
558     \MT@ifdim\MT@size<{#1}\relax{%
559       \MT@ifdim\MT@size<{#2}%
560       \MT@inlist@true
561       \relax
562     }%
563   }%
564   \ifMT@inlist@
565     \def\MT@size@name{#3}%
566     \expandafter\MT@tlist@break
567   \fi
568 }

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

\MT@repeat 569 /package
570 \def\MT@loop#1\MT@repeat{%
571   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
572   \MT@iterate \let\MT@iterate\relax
573 }
574 \let\MT@repeat\fi

```

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

```
575 \def\MT@while@num#1#2#3{%
576   \@tempcnta#1\relax
577   \MT@loop #3%
578   \advance\@tempcnta \@ne
579   \ifnum\@tempcnta < #2\MT@repeat
580 }
```

`\MT@do@font`      Execute  $\langle \#1 \rangle$  256 times.

```
581 \def\MT@do@font{\MT@while@num\z@\cc@lvi}
582 <package>
```

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

`\MT@increment`

```
583 \newcount\MT@count
584 \def\MT@increment#1{%
585   ^^X \edef#1{\number\numexpr #1 + 1\relax}%
586   ^^Q \MT@count=#1\relax
587   ^^Q \advance\MT@count \@ne
588   ^^Q \edef#1{\number\MT@count}%
589 }
```

`\MT@scale`      Multiply and divide a counter. If we are using e- $\text{\TeX}$ , 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.

```
590 \def\MT@scale#1#2#3{%
591   ^^Q \multiply #1 #2\relax
592   \ifnum #3 = \z@
593     ^^X #1=\numexpr #1 * #2\relax
594   \else
595     ^^X #1=\numexpr #1 * #2 / #3\relax
596     ^^Q \divide #1 #3\relax
597   \fi
598 }
```

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

`\MT@abbr@ex`

```
599 \def\MT@abbr@pr{protrusion}
600 \def\MT@abbr@ex{expansion}
601 \def\MT@abbr@pr@c{protrusion codes}
602 \def\MT@abbr@ex@c{expansion codes}
603 \def\MT@abbr@pr@inh{protrusion inheritance}
604 \def\MT@abbr@ex@inh{expansion inheritance}
605 \def\MT@abbr@nl{noligatures}
606 \def\MT@abbr@sp{spacing}
607 \def\MT@abbr@sp@c{interword spacing codes}
608 \def\MT@abbr@sp@inh{interword spacing inheritance}
609 \def\MT@abbr@kn{kerning}
610 \def\MT@abbr@kn@c{kerning codes}
611 \def\MT@abbr@kn@inh{kerning inheritance}
612 \def\MT@abbr@tr{tracking}
613 \def\MT@abbr@tr@c{tracking amount}
```

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

`\MT@rbba@protrusion`

`\MT@rbba@expansion`

`\MT@rbba@spacing`

`\MT@rbba@kerning`

`\MT@rbba@tracking`

```
614 \def\MT@rbba@protrusion{pr}
615 \def\MT@rbba@expansion{ex}
616 \def\MT@rbba@spacing{sp}
617 \def\MT@rbba@kerning{kn}
618 \def\MT@rbba@tracking{tr}
```

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

`\MT@features@long` 619 `\def\MT@features{pr,ex,sp,kn,tr}`  
620 `\def\MT@features@long{protrusion,expansion,spacing,kerning,tracking}`

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

621 `\def\MT@is@feature#1{%`  
622 `\MT@exp@one@n\MT@in@clist\@tempa\MT@features@long`  
623 `\ifMT@inlist@`  
624 `\expandafter\@firstofone`  
625 `\else`  
626 `\MT@error{'\@tempa' is not an available micro-typographic\MessageBreak`  
627 `feature. Ignoring #1}{Available features are: '\MT@features@long'.}%`  
628 `\expandafter\@gobble`  
629 `\fi`  
630 `}`

### 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 14.4.5)

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.

631 `\@ifl@aded{tex}{wordcount}{%`  
632 `\MT@warning@n{Detected the 'wordcount' utility.\MessageBreak`  
633 `Disabling '\MT@MT', since it wouldn't work}%`  
634 `\MT@clear@options\MT@restore@catcodes\endinput}\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.

635 `\</package>`  
636 `\<plain>\MT@requires@latex1{`  
637 `\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.

638 `\def\MT@addto@setup{\g@addto@macro\MT@setup@`

It will be executed at the end of the preamble, and emptied (the combine class calls it repeatedly).

639 `\AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}`

Don’t hesitate with `miniltx`.

640 `\<plain>}{\let\MT@addto@setup\@firstofone}`

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

641 `\def\MT@with@package@T#1{\ifpackageloaded{#1}\@firstofone\@gobble}`



642 *(\*package)*

`\MT@with@babel@and@T`      $\text{\LaTeX}$ 's `\@ifpackagewith` ignores the class options.

```
643 \def\MT@with@babel@and@T#1{%
644   \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
645     \@expandtwoargs\MT@in@clist{#1}
646     {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
647     \ifMT@inlist@expandafter\@gobble\fi
648   }@gobble
649 }
```

Don't load `letterspace`.

```
650 \MT@let@nc{ver@letterspace.sty}\@empty
```

`\MT@ledmac@setup`     The `ledmac` package first saves each paragraph in a box, from which it then splits  
`\MT@led@unhbox@line`     off the lines one by one. This will destroy character protrusion. (There aren't any  
`\MT@led@kern`     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 pdf $\text{\TeX}$  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 pdf $\text{\TeX}$  1.21b (aka. 1.30.0).

```
651 \MT@requires@pdftex5{
652   \def\MT@ledmac@setup{%
653     \ifMT@protrusion
654       \MT@ifdefined@c@TF\l@dunhbox@line{%
655         \MT@info@nl{Patching ledmac to enable character protrusion}%
656         \newdimen\MT@led@kern
657         \let\MT@led@unhbox@line\l@dunhbox@line
658         \renewcommand*{\l@dunhbox@line}[1]{%
659           \ifhbox##1%
660             \MT@led@kern=\rightmarginkern##1%
661             \kern\leftmarginkern##1%
662             \MT@led@unhbox@line##1%
663             \kern\MT@led@kern
664           \fi
665         }%
666       }%
667       \MT@warning@nl{%
668         Character protrusion in paragraphs with line\MessageBreak
669         numbering will only work if you update ledmac}%
670     }%
671   \fi
672 }
673 }{
674   \def\MT@ledmac@setup{%
675     \ifMT@protrusion
676       \MT@warning@nl{%
677         The pdftex version you are using does not allow\MessageBreak
678         character protrusion in paragraphs with line\MessageBreak
679         numbering by the 'ledmac' package.\MessageBreak
680         Upgrade pdftex to version 1.30 or later}%
681     \fi
682   }
683 }
```

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

```
684 \def\MT@restore@p@h{\chardef\%~\% \chardef\#~\# }
```

`\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`.)

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

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

```
686 \MT@if@false
687 \MT@with@babel@and@T{spanish} \MT@if@true
688 \MT@with@babel@and@T{galician} \MT@if@true
689 \MT@with@babel@and@T{mexican} \MT@if@true
690 \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.)

```
691 \MT@with@package@T{csquotes}{%
692 \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`.

```
693 \MT@if@false
694 \MT@with@package@T{hyperref} \MT@if@true
695 \MT@with@package@T{tex4ht} \MT@if@true
696 \ifMT@if@MT@restore@p@h\fi
697 }
```

Check again at the end of the preamble.

```
698 
```

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

```
701 \MT@with@package@T{pdfcpot}{%
702 \MT@error{Detected the `pdfcpot' package!\MessageBreak
703 `~\MT@MT' and `pdfcpot' may not be used together}{%
704 The `pdfcpot' package provides an interface to character protrusion.\MessageBreak
705 So does the `~\MT@MT' package. Using both packages at the same\MessageBreak
706 time will almost certainly lead to undesired results. Have your choice!}%
707 }
708 \MT@with@package@T{ledmac}\MT@ledmac@setup
```

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

```
709 \let\MT@setupfont@hook\empty
710 \MT@if@false
711 \MT@with@babel@and@T{spanish} \MT@if@true
712 \MT@with@babel@and@T{galician} \MT@if@true
713 \MT@with@babel@and@T{mexican} \MT@if@true
714 \ifMT@if@
715 \g@addto@macro\MT@setupfont@hook{%
716 \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
717 \fi
718 \MT@with@package@T{csquotes}{%
719 \ifpackage@later{csquotes}{2005/05/11}{%
```

```

720 \g@addto@macro\MT@setupfont@hook\@disablequotes
721 }{%
722 \MT@warning@nl{%
723   Should you receive warnings about unknown slot\MessageBreak
724   numbers, try upgrading the `csquotes' package}%
725 }%
726 }

```

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.

```

727 \MT@if@false
728 </package>
729 <plain> \MT@requires@latex2{
730   \MT@with@package@T{hyperref}{%
731     \pdfstringdefDisableCommands{%
732 <*package>
733       \let\pickup@font\MT@orig@pickupfont
734       \let\textmicrotypecontext\@secondoftwo
735       \let\microtypecontext\@gobble
736 </package>
737       \def\lststyle{\pdfstringdefWarn\lststyle}%
738       \def\textls#1#\pdfstringdefWarn\textls}%
739     }%
740 <package> \MT@if@true
741 }
742 <plain> }\relax
743 <*package>
744 \MT@with@package@T{tex4ht}\MT@if@true
745 \ifMT@if@g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi

```

The `listings` package makes numbers and letters active,

```

746 \MT@with@package@T{listings}{%
747   \g@addto@macro\MT@cfg@catcodes{%
748     \MT@while@num{"30}{ "3A}{\catcode\@tempcnta 12\relax}%
749     \MT@while@num{"41}{ "5B}{\catcode\@tempcnta 11\relax}%
750     \MT@while@num{"61}{ "7B}{\catcode\@tempcnta 11\relax}%
751   }%

```

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

```

752 \g@addto@macro\MT@setupfont@hook{%
753   \catcode`\z@

```

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.

```

754 \let\lst@ProcessLetter\empty
755 }%
756 }

```

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.

```

757 </package>
758 <plain> \MT@requires@latex2{
759   \MT@with@package@T{soul}{%
760     \soulregister\lststyle 0%
761     \soulregister\textls 1%
762   }

```

Under plain TeX, 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.

```

763 <plain>
764   {\ifx\Soul@\undefined\else
765     \soulregister\lsstyle 0%
766     \soulregister\textls 1%
767     \fi}
768 </plain>
769 <package>

```

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, \py@macron had only one argument.

```

770 \MT@with@package@T{pinyin}{%
771   \let\MT@orig@py@macron\py@macron
772   \ifpackageafter{pinyin}{2006/10/17}{% 4.7.0
773     \def\py@macron#1#2{%
774       \let\pickup@font\MT@orig@pickupfont
775       \MT@orig@py@macron{#1}{#2}%
776       \let\pickup@font\MT@pickupfont}%
777     }{%
778       \def\py@macron#1{%
779         \let\pickup@font\MT@orig@pickupfont
780         \MT@orig@py@macron{#1}%
781         \let\pickup@font\MT@pickupfont}%
782       }%
783     }
784 </package>
785 }
786 <package>

```

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

```

787 \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.

```

788 \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).

```

789 \MT@requires@pdftex7
790 {\g@addto@macro\MT@setupfont\MT@copy@font}\relax

```

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

```

791 \g@addto@macro\MT@setupfont{%
792   \MT@exp@two@c\MT@split@name\string\MT@font/\@nil

```

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

```

793   \MT@exp@one@n\MT@find@file\MT@family
794   \ifx\MT@familyalias\@empty \else
795     \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.)

```
796 % \ifx\fontencoding\cfontencoding\else\@enc@update\fi
797 }
```

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

```
798 \MT@requires@pdftex6
799 {\g@addto@macro\MT@setupfont\MT@tracking}\relax
800 \g@addto@macro\MT@setupfont{%
801   \MT@check@font
802   \ifMT@inlist@
803   <debug>\MT@showpdfannot2%
804   \else
805   \MT@vinfo{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.

```
806 \MT@protrusion
807 \MT@expansion
808 }
```

Interword spacing and kerning (pdfTeX 1.40).

```
809 \MT@requires@pdftex6
810 {\g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}}\relax
```

Disable ligatures (pdfTeX 1.30).

```
811 \MT@requires@pdftex5
812 {\g@addto@macro\MT@setupfont\MT@noligatures}\relax
813 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
814 <debug>\MT@showpdfannot1%
```

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

```
815 \MT@register@font
816 \fi
817 }
```

`\MT@copy@font`      The new `\pdfcopyfont` command allows to expand 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.

```
818 \let\MT@copy@font\relax
819 \MT@requires@pdftex7{
820 \def\MT@copy@font@{%
```

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

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

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

```
822 \expandafter\ifx\MT@font@copy\relax
823 \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
824 \expandafter\ifx\MT@font@orig\relax
825 \MT@exp@two@c\MT@glet\MT@font@orig\font@name
```

```

826 \else
827 \MT@exp@two@c\let\font@name\MT@font@orig
828 \fi
829 \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
830 <debug>\MT@dinfol{creating new copy: \MT@font@copy}%

```

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

```

831 \MT@map@clist@c\MT@active@features{%
832 \MT@exp@cs@ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
833 \def\@tempa{##1}%
834 \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@rem@from@list
835 \fi
836 }%
837 \fi
838 \MT@exp@two@c\let\MT@font\MT@font@copy

```

We only need the font identifier for letterspacing.

```

839 \let\font@name\MT@font@copy

```

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

```

840 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
841 }

```

\MT@rem@from@list

```

842 \def\MT@rem@from@list#1{%
843 \MT@exp@cs@ifx{MT@\@tempa @#1font@list}\relax\else
844 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
845 \MT@font \csname MT@\@tempa @#1font@list\endcsname
846 \fi
847 }
848 }\relax

```

*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  
 \MT@encoding  
 \MT@family  
 \MT@series  
 \MT@shape  
 \MT@size

Split up the font name (`(/6)` may be a protrusion/expansion context and/or a letterspacing amount).

```

849 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
850   \def\MT@encoding{#1}%
851   \def\MT@family   {#2}%
852   \def\MT@series   {#3}%
853   \def\MT@shape    {#4}%
854   \def\MT@size     {#5}%
\MT@familyalias  Alias family?
855   \MT@ifdefined@n@TF{MT@\MT@family @alias}%
856   {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
857   {\let\MT@familyalias\@empty}%
858 }

\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 859 \newif\ifMT@do
860 \def\MT@maybe@do#1{%
    (but only if the feature isn't globally set to false)
861   \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).

862   \MT@dotrue
863   \edef\@tempa{\csname MT@#1\setname\endcsname}%
864   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
865     \MT@ifdefined@n@TF{MT@checklist@#1}%
866     {\csname MT@checklist@#1\endcsname}%
867     {\MT@checklist@{#1}}%
868     {#1}%
869   }%
870   \else
871     \MT@dofalse
872   \fi
873   \ifMT@do

    \MT@feat stores the current feature.

874   \def\MT@feat{#1}%
875   \csname MT@set@#1\codes\endcsname
876   \else
877     \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
878   \fi
879 }

\MT@dinfo@list
880 <debug>\def\MT@dinfo@list#1#2#3{\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#1}:
881 <debug> #2 \ifx\#3\list empty\else \@nameuse{MT@#2}' #3 list\fi}%
882 <debug>}}

\MT@checklist@  The generic test (<#1> is the axis, <#2> the feature, \@tempa contains the set name).
883 \def\MT@checklist@#1#2{%
884 <!debug> \MT@ifdefined@n@T
885 <debug> \MT@ifdefined@n@TF
886   {MT@#2list@#1@\@tempa}{%

    Begin an \expandafter orgy to test whether the font attribute is in the list.

887   \expandafter\MT@exp@one@n\expandafter\MT@in@clist
888   \csname MT@#1\expandafter\endcsname
889   \csname MT@#2list@#1@\@tempa\endcsname
890   \ifMT@inlist@

```

```

891 <debug>\MT@info@list{#2}{#1}{in}%
892     \MT@dotrue
893     \else
894 <debug>\MT@info@list{#2}{#1}{not in}%
895     \MT@dofalse
896     \expandafter\MT@clist@break
897     \fi
898 }%

```

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.

```

899 <debug> {\MT@info@list{#2}{#1}{}}%
900 }

```

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

```

901 \def\MT@checklist@family#1{%
902 <!debug> \MT@ifdefined@n@T
903 <debug> \MT@ifdefined@n@TF
904     {MT@#1list@family@ \@tempa}%
905     \MT@exp@two@n\MT@in@clist
906     \MT@family{\csname MT@#1list@family@ \@tempa\endcsname}%
907     \ifMT@inlist@
908 <debug>\MT@info@list{#1}{family}{in}%
909     \MT@dotrue
910     \else
911 <debug>\MT@info@list{#1}{family}{not in}%
912     \MT@dofalse
913     \ifx\MT@familyalias\@empty \else
914     \MT@exp@two@n\MT@in@clist
915     \MT@familyalias{\csname MT@#1list@family@ \@tempa\endcsname}%
916     \ifMT@inlist@
917 <debug> \MT@info@list{#1}{family alias}{in}%
918     \MT@dotrue
919 <debug>\else\MT@info@list{#1}{family alias}{not in}%
920     \fi
921     \fi
922     \fi
923     \ifMT@do \else
924     \expandafter\MT@clist@break
925     \fi
926 }%
927 <debug> {\MT@info@list{#1}{family}{}}%
928 }

```

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

```

929 \def\MT@checklist@size#1{%
930 <!debug> \MT@ifdefined@n@T
931 <debug> \MT@ifdefined@n@TF
932     {MT@#1list@size@ \@tempa}%
933     \MT@exp@cs\MT@in@rlist{MT@#1list@size@ \@tempa}%
934     \ifMT@inlist@
935 <debug>\MT@info@list{#1}{size}{in}%
936     \MT@dotrue
937     \else
938 <debug>\MT@info@list{#1}{size}{not in}%
939     \MT@dofalse
940     \expandafter\MT@clist@break
941     \fi
942 }%
943 <debug> {\MT@info@list{#1}{size}{}}%
944 }

```



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

```

945 \def\MT@checklist@font#1{%
946   <!--debug--> \MT@ifdefined@n@T
947   <!--debug--> \MT@ifdefined@n@TF
948   {MT@#1list@font@\@tempa}{%

```

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

```

949   \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
950   \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
951   \@tempb \csname MT@#1list@font@ \@tempa\endcsname
952   \ifMT@inlist@
953   <!--debug--> \MT@info@list{#1}{font}{in}%
954   \expandafter\MT@clist@break
955   \else
956   <!--debug--> \MT@info@list{#1}{font}{not in}%
957   \MT@dofalse
958   \fi
959   }%
960   <!--debug--> {\MT@info@list{#1}{font}}}%
961 }

```

### 14.2.1 Protrusion

`\MT@protrusion` Set up for protrusion?

```

962 \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.

```

963 \def\MT@set@pr@codes{%

```

Check whether and if, which list should be applied to the current font.

```

964   \MT@if@list@exists{%
965   \MT@get@font@dimen@six
966   \MT@get@opt
967   \MT@reset@pr@codes

```

Get the name of the inheritance list and parse it.

```

968   \MT@get@inh@list

```

Set an input encoding?

```

969   \MT@set@inputenc{c}%

```

Load additional lists?

```

970   \MT@load@list\MT@pr@c@name
971   \MT@set@listname

```

Load the main list.

```

972   \MT@let@cn\@tempc{\MT@pr@c@\MT@pr@c@name}%
973   \expandafter\MT@set@codes\@tempc,\relax,%
974   }\MT@reset@pr@codes
975 }

```

`\MT@gobble@settings`

```

976 \def\MT@gobble@settings#1\@tempc,\relax,{}

```

`\MT@get@font@dimen@six` If `\fontdimen6` is zero, character protrusion won't work, and we can skip the settings (for example, the dsfont fonts don't specify this dimension; this is probably a bug).

`\MT@dimen@six`

```

977 \def\MT@get@font@dimen@six{%
978   \ifnum\fontdimen6\MT@font=\z@
979     \MT@warning@n1{%
980       Font '\MT@font' does not specify its\MessageBreak
981       \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
982       \@nameuse{MT@abbr@\MT@feat} will not work with this font}%
983     \expandafter\MT@gobble@settings
984   \else
985     \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
986   \fi
987 }

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

988 \def\MT@set@all@pr#1#2{%
989   debug\MT@info@n1{3}{-- lp/rp: setting all to #1/#2}%
990   \let\MT@temp\@empty
991   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lpcode\MT@font\@tempcnta=#1\relax}}%
992   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rpcode\MT@font\@tempcnta=#2\relax}}%
993   \MT@do@font\MT@temp
994 }

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

995 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
996 \let\MT@reset@pr@codes\relax

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

997 \def\MT@the@pr@code{\@tempcntb}
998 \MT@requires@pdftex6{
999   \def\MT@the@pr@code@tr{%
1000     \numexpr\@tempcntb+\MT@letterspace@/2\relax
1001   }
1002 } \relax

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

1003 \def\MT@set@codes#1,{%
1004   \ifx\relax#1\@empty\else
1005     \MT@split@codes #1==\relax
1006     \expandafter\MT@set@codes
1007   \fi
1008 }

\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.

1009 \def\MT@split@codes#1=#2=#3\relax{%
1010   \def\@tempa{#1}%
1011   \ifx\@tempa\@empty \else
1012     \MT@get@slot
1013     \ifnum\MT@char > \m@ne
1014       \MT@get@char@unit
1015       \csname MT@\MT@feat @split@val\endcsname#2\relax
1016     \fi
1017   \fi
1018 }

\MT@pr@split@val

1019 \def\MT@pr@split@val#1,#2\relax{%
1020   \def\@tempb{#1}%

```

```

1021 \MT@ifempty\@tempb\relax{%
1022   \MT@scale@to@em
1023   \lcode\MT@font\MT@char=\MT@the@pr@code
1024 (debug)\MT@info{n1}{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char: [#1]}%
1025 }%
1026 \def\@tempb{#2}%
1027 \MT@ifempty\@tempb\relax{%
1028   \MT@scale@to@em
1029   \rcode\MT@font\MT@char=\MT@the@pr@code
1030 (debug)\MT@info{n1}{4}{;;; rp (\MT@char): \number\rcode\MT@font\MT@char: [#2]}%
1031 }%

```

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

```

1032 \MT@ifdefined@c@T\MT@pr@inh@name{%
1033   \MT@ifdefined@nT{\MT@inh@\MT@pr@inh@name @\MT@char @}{%
1034     \MT@exp@cs\MT@map@tlist@c
1035     {\MT@inh@\MT@pr@inh@name @\MT@char @}%
1036     \MT@set@pr@heirs
1037   }%
1038 }%
1039 }

```

`\MT@scale@to@em`

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

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

```

1040 \MT@requires@pdftex3{
1041   \def\MT@scale@to@em{%
1042     \@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`.

```

1043   \MT@scale\@tempcntb \@tempb \MT@dimen@six
1044   \ifnum\@tempcntb=\z@ \else
1045     \MT@scale@factor
1046   \fi
1047 }

```

`\MT@get@charwd`

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

```

1048 \def\MT@get@charwd{%
1049   ^^X \MT@count=\fontcharwd\MT@font\MT@char\relax
1050   ^^Q \setbox\z@=\hbox{\MT@font \char\MT@char}%
1051   ^^Q \MT@count=\wd\z@
1052   \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1053 }

```

For letterspaced fonts, we have to subtract the letterspacing amount from the characters’ widths. The protrusion amounts will be adjusted in `\MT@set@pr@codes`.

The letterspaced font is already loaded so that 1 em = \fontdimen 6.

```

1054 \MT@requires@pdftex6{
1055   \g@addto@macro\MT@get@charwd{%
1056     \MT@ifdefined@c@T\MT@letterspace@
1057     {\advance\MT@count-\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1058   }
1059 } \relax
1060 }{

```

No adjustment with versions 0.14f and 0.14g.

```

1061 \def\MT@scale@to@em{%
1062   \MT@count=\@tempb\relax
1063   \ifnum\MT@count=z@ \else
1064     \MT@scale@factor
1065   \fi
1066 }

```

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

```

1067 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1068 }

```

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

```

1069 \def\MT@get@font@dimen#1{%
1070   \ifnum\fontdimen#1\MT@font=z@
1071     \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1072       \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1073       You should use a different 'unit' for \MT@curr@list@name}%
1074   \else
1075     \MT@count=\fontdimen#1\MT@font
1076   \fi
1077 }

```

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

```

1078 \def\MT@info@missing@char{%
1079   \MT@info@n1{Character '\the\MT@toks'
1080     ^^X \iffontchar\MT@font\MT@char
1081       has a width of 0pt
1082     ^^X \else is missing\fi
1083     ^^Q \MessageBreak (it's probably missing)
1084     \MessageBreak in font '\MT@font'. \MessageBreak
1085     Ignoring protrusion settings for this character}%
1086 }

```

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

```

1087 \def\MT@scale@factor{%
1088   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1089     \expandafter\MT@scale\expandafter \@tempcntb
1090     \csname MT@\MT@feat @factor@\endcsname \@m
1091   \fi
1092   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
1093     \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1094   \else
1095     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1096       \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1097     \fi
1098   \fi
1099 }

```

\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.

```

1100 \def\MT@warn@code@too@large#1{%
1101   \@tempcnta=#1\relax
1102   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1103     \expandafter\MT@scale\expandafter\@tempcnta\expandafter
1104     \@m \csname MT@\MT@feat @factor@\endcsname
1105   \fi
1106   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1107   \MT@warning@n1{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
1108     is too large for character\MessageBreak
1109     \the\MT@toks' in \MT@curr@list@name.\MessageBreak
1110     Setting it to the maximum of \number\@tempcnta}%
1111   \@tempcntb=#1\relax
1112 }

```

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

```

1113 \def\MT@get@opt{%
1114   \MT@set@listname

```

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

```

\MT@sp@factor@ 1115 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 1116   \MT@let@nn{MT@\MT@feat @factor@}%
1117   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1118   \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1119     \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1120   }{%
1121     \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1122   }%

```

`\MT@pr@unit@` The unit can only be evaluated here, since it might be font-specific. If it's `\@empty`, it's relative to character widths, if it's `-1`, relative to space dimensions.

```

\MT@sp@unit@ 1123 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
\MT@kn@unit@ 1124   \MT@let@nn{MT@\MT@feat @unit@}%
1125   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1126   \MT@exp@cs@ifx{MT@\MT@feat @unit@}\@empty
1127   \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1128     relative to character widths}%
1129   \else
1130     \MT@exp@cs@ifx{MT@\MT@feat @unit@}\m@ne
1131     \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1132       relative to width of space}%
1133   \fi
1134   \fi
1135   }{%
1136     \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
1137   }%

```

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

```

1138 \let\MT@get@char@unit\relax
1139 \let\MT@get@space@unit\@gobble
1140 \MT@exp@cs@ifx{MT@\MT@feat @unit@}\@empty
1141 \let\MT@get@char@unit\MT@get@charwd
1142 \else
1143   \MT@exp@cs@ifx{MT@\MT@feat @unit@}\m@ne
1144   \let\MT@get@space@unit\MT@get@font@dimen
1145   \else
1146     \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}%
1147   \fi
1148 \fi

```

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

```

1149 \MT@ifdefined@n@T{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @preset}{%
1150   \csname MT@preset@MT@feat\endcsname
1151   \MT@let@nc{MT@reset@MT@feat @codes}\relax
1152 }%
1153 }

```

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

```

1154 \def\MT@get@unit#1{%
1155   \expandafter\MT@get@unit@#1 e!\@nil
1156   \ifx\x\@empty\else\let#1\x\fi
1157   \@defaultunits\@tempdima#1 pt\relax\@nnil
1158   \ifdim\@tempdima=\z@
1159     \MT@warning@n1{%
1160       Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
1161       width. Setting factors of list \@nameuse{MT@MT@feat @c@name}'\MessageBreak
1162       relative to character widths instead}%
1163     \let#1\@empty
1164     \let\MT@get@char@unit\MT@get@charwd
1165   \else
1166     \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
1167               to \the\@tempdima}%
1168     \MT@count=\@tempdima\relax
1169   \fi
1170 }
1171 \def\MT@get@unit@#1e#2#3\@nil{%
1172   \ifx\#3\\\let\x\@empty \else
1173     \if m#2%
1174       \edef\x{#1\fontdimen6\MT@font}%
1175     \else
1176       \if x#2%
1177         \edef\x{#1\fontdimen5\MT@font}%
1178       \fi
1179     \fi
1180   \fi
1181 }

```

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

```

1182 \def\MT@set@inputenc#1{%

```

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

```

1183   \def\MT@cat{#1}%
1184   \edef\@tempa{MT@MT@feat @#1@csname MT@MT@feat @#1@name\endcsname @inputenc}%
1185   \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1186 }

```

\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.

```

1187 \MT@addto@setup{%
1188   \ifpackageloaded{inputenc}{
1189     \ifpackageafter{inputenc}{2006/02/22}{
1190       \def\MT@set@inputenc@{%
1191         \MT@ifstreq\inputencodingname{csname\@tempa\endcsname}\relax
1192         \MT@load@inputenc
1193       }
1194     }{
1195       \let\MT@set@inputenc@\MT@load@inputenc

```

```

1196     }
1197   }{
1198     \def\MT@set@inputenc{%
1199       \MT@warning@n1{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1200         \MessageBreak package isn't loaded. Ignoring input encoding}%
1201     }
1202   }
1203 }

```

`\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.

```

1204 \def\MT@load@inputenc{%
1205   \MT@cfg@catcodes
1206   <debug>\MT@info@n1{1}{loading input encoding: \@nameuse{\@tempa}}%
1207   \inputencoding{\@nameuse{\@tempa}}%
1208 }

```

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

```

1209 \def\MT@set@pr@heirs#1{%
1210   \lcode\MT@font#1=\lcode\MT@font\MT@char
1211   \rcode\MT@font#1=\rcode\MT@font\MT@char
1212   <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1213   <debug>\MT@info@n1{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
1214   <debug>                                     \number\rcode\MT@font\MT@char}%
1215 }

```

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

```

\MT@preset@pr@ 1216 \def\MT@preset@pr{%
1217   \expandafter\expandafter\expandafter\MT@preset@pr@
1218   \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1219 }
1220 \def\MT@preset@pr@#1,#2\@nil{%
1221   \ifx\MT@pr@unit@\empty
1222     \MT@warn@preset@tewidth{pr}%
1223     \let\MT@preset@aux\MT@preset@aux@factor
1224   \else
1225     \def\MT@preset@aux{\MT@preset@aux@space2}%
1226   \fi
1227   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1228   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1229   \MT@set@all@pr@\@tempa\@tempb
1230 }

```

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

```

\MT@preset@aux@factor 1231 \def\MT@preset@aux@factor#1#2{%
1232   \@tempcntb=#1\relax
\MT@preset@aux@space 1233   \MT@scale@factor
1234   \edef#2{\number\@tempcntb}%
1235 }
1236 \def\MT@preset@aux@space#1#2#3{%
1237   \def\@tempb{#2}%
1238   \MT@get@space@unit#1%
1239   \MT@scale@to@em
1240   \edef#3{\number\@tempcntb}%
1241 }

```

`\MT@warn@preset@tewidth`

```

1242 \def\MT@warn@preset@tewidth#1{%
1243   \MT@warning@n1{%
1244     Cannot preset characters relative to their widths\MessageBreak
1245     for \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1246     \MessageBreak relative to lem instead}%

```

1247 }

### 14.2.2 Expansion

`\MT@expansion` Set up for expansion?

1248 `\def\MT@expansion{\MT@maybe@do{ex}}`

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

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

```
1249 \def\MT@set@ex@codes@{%
1250   \MT@if@list@exists{%
1251     \MT@get@ex@opt
1252     \let\MT@get@char@unit\relax
1253     \MT@reset@ef@codes
1254     \MT@get@inh@list
1255     \MT@set@inputenc{c}%
1256     \MT@load@list\MT@ex@code@name
1257     \MT@set@listname
1258     \MT@let@cn\@tempc{\MT@ex@code@\MT@ex@code@name}%
1259     \expandafter\MT@set@codes\@tempc,\relax,%
1260     \MT@expandfont
1261   }\relax
1262 }
```

`\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.

```
1263 \newif\ifMT@nonselected
1264 \def\MT@set@ex@codes@n{%
1265   \MT@nonselectedtrue
1266   \MT@if@list@exists
1267   \MT@get@ex@opt
1268   {%
1269     \let\MT@stretch@ \MT@stretch
1270     \let\MT@shrink@ \MT@shrink
1271     \let\MT@step@ \MT@step
1272     \let\MT@auto@ \MT@auto
1273     \let\MT@ex@factor@ \MT@ex@factor
1274   }%
1275   \MT@reset@ef@codes
1276   \MT@expandfont
1277   \MT@nonselectedfalse
1278 }
```

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

1279 `\let\MT@set@ex@codes\MT@set@ex@codes@n`

`\MT@expandfont` Expand the font.

```
1280 \def\MT@expandfont{%
1281   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1282 }
```

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

1283 `\def\MT@set@all@ex#1{%`



```

1284 (debug)\MT@info{n1}{3}{-- ex: setting all to \number#1}%
1285 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1286 }
1287 \def\MT@reset@ef@codes@{\MT@set@all@ex\MT@ex@factor@}

```

\MT@reset@ef@codes      However, this is only necessary for versions prior to 1.20.

```

1288 \MT@requires@pdfTeX4{
1289 \def\MT@reset@ef@codes{%
1290 \ifnum\MT@ex@factor@=\@m \else
1291 \MT@reset@ef@codes@
1292 \fi
1293 }
1294 }{
1295 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1296 }

```

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

```

1297 \def\MT@ex@split@val#1\relax{%
1298 \@tempcntb=#1\relax

```

Take an optional factor into account.

```

1299 \ifnum\MT@ex@factor@=\@m \else
1300 \MT@scale\@tempcntb \MT@ex@factor@ \@m
1301 \fi
1302 \ifnum\@tempcntb > \MT@ex@max
1303 \MT@warn@ex@too@large\MT@ex@max
1304 \else
1305 \ifnum\@tempcntb < \MT@ex@min
1306 \MT@warn@ex@too@large\MT@ex@min
1307 \fi
1308 \fi
1309 \efcode\MT@font\MT@char=\@tempcntb
1310 (debug)\MT@info{n1}{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%

```

Heirs, heirs, I love thy heirs.

```

1311 \MT@ifdefined@c@T\MT@ex@inh@name{%
1312 \MT@ifdefined@n@T{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1313 \MT@exp@cs\MT@map@tlist@c{\MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
1314 }%
1315 }%
1316 }

```

\MT@warn@ex@too@large

```

1317 \def\MT@warn@ex@too@large#1{%
1318 \MT@warning{n1}{Expansion factor \number\@tempcntb\space too large for
1319 character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1320 Setting it to the maximum of \number#1}%
1321 \@tempcntb=#1\relax
1322 }

```

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

```

\MT@ex@factor@ 1323 \def\MT@get@ex@opt{%
\MT@stretch@ 1324 \MT@set@listname
1325 \MT@ifdefined@n@TF{\MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1326 \MT@let@cn\MT@ex@factor@{\MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1327 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
1328 }{%
\MT@auto@ 1329 \let\MT@ex@factor@\MT@ex@factor
1330 }%
1331 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1332 \MT@get@ex@opt@{shrink}{Setting shrink limit to \number\MT@shrink@}%
1333 \MT@get@ex@opt@{step}{Setting expansion step to \number\MT@step@}%

```

```

1334 \def\@tempa{autoexpand}%
1335 \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1336 \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @preset}}}%
1337 \MT@preset@ex
1338 \let\MT@reset@ef@codes\relax
1339 }%
1340 }

```

\MT@get@ex@opt@

```

1341 \def\MT@get@ex@opt@#1#2{%
1342 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}}}%
1343 \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
1344 \MT@vinfo{... : #2}%
1345 }{%
1346 \MT@let@nn{MT@#1@}{MT@#1}%
1347 }%
1348 }

```

\MT@set@ex@heirs

```

1349 \def\MT@set@ex@heirs#1{%
1350 \efcode\MT@font#1=\efcode\MT@font\MT@char
1351 <debug>\MT@info@nl{2}{-- heir of \MT@char: #1}%
1352 <debug>\MT@info@nl{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1353 }

```

\MT@preset@ex

```

1354 \def\MT@preset@ex{%
1355 \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
1356 \MT@scale@factor
1357 \MT@set@all@ex@tempcntb
1358 }

```

### 14.2.3 Interword spacing (glue)

\MT@spacing Adjustment of interword spacing?

```

1359 \MT@requires@pdftex6{
1360 \def\MT@spacing{\MT@maybe@do{sp}}

```

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

```

1361 \def\MT@set@sp@codes{%
1362 \MT@if@list@exists{%
1363 \MT@get@font@dimen@six
1364 \MT@get@opt
1365 \MT@reset@sp@codes
1366 \MT@get@inh@list
1367 \MT@set@inputenc{c}%
1368 \MT@load@list\MT@sp@c@name
1369 \MT@set@listname
1370 \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
1371 \expandafter\MT@set@codes\@tempc,\relax,%
1372 }\MT@reset@sp@codes
1373 }

```

\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).

```

1374 \def\MT@sp@split@val#1,#2,#3\relax{%
1375 \def\@tempb{#1}%
1376 \MT@ifempty\@tempb\relax{%
1377 \MT@get@space@unit2%
1378 \MT@scale@to@em

```

```

1379 \kbscode\MT@font\MT@char=\@tempcntb
1380 <debug>\MT@info@n1{4}{;;; knbs (\MT@char): \number\kbscode\MT@font\MT@char: [#1]}%
1381 }%
1382 \def\@tempb{#2}%
1383 \MT@ifempty\@tempb\relax{%
1384 \MT@get@space@unit3%
1385 \MT@scale@to@em
1386 \stbscode\MT@font\MT@char=\@tempcntb
1387 <debug>\MT@info@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1388 }%
1389 \def\@tempb{#3}%
1390 \MT@ifempty\@tempb\relax{%
1391 \MT@get@space@unit4%
1392 \MT@scale@to@em
1393 \shbscode\MT@font\MT@char=\@tempcntb
1394 <debug>\MT@info@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1395 }%
1396 \MT@ifdefined@c@T\MT@sp@inh@name{%
1397 \MT@ifdefined@n@T\MT@inh@MT@sp@inh@name @\MT@char @}%
1398 \MT@exp@cs\MT@map@tlist@c\MT@inh@MT@sp@inh@name @\MT@char @\MT@set@sp@heirs
1399 }%
1400 }%
1401 }

```

\MT@set@sp@heirs

```

1402 \def\MT@set@sp@heirs#1{%
1403 \kbscode\MT@font#1=\kbscode\MT@font\MT@char
1404 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1405 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1406 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1407 <debug>\MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\kbscode\MT@font\MT@char/%
1408 <debug> \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1409 }

```

\MT@set@all@sp

```

\MT@reset@sp@codes 1410 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1411 <debug>\MT@info@n1{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1412 \let\MT@temp\@empty
1413 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\kbscode\MT@font\@tempcnta=#1\relax}}%
1414 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1415 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1416 \MT@do@font\MT@temp
1417 }
1418 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1419 \let\MT@reset@sp@codes\relax

```

\MT@preset@sp

```

\MT@preset@sp@ 1420 \def\MT@preset@sp{%
1421 \expandafter\expandafter\expandafter\MT@preset@sp@
1422 \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1423 }
1424 \def\MT@preset@sp@#1,#2,#3\@nil{%
1425 \ifx\MT@sp@unit@\@empty
1426 \MT@warn@preset@to@width{sp}%
1427 \MT@ifempty{#1}\let\@tempa\@empty{\MT@preset@aux@factor{#1}\@tempa}%
1428 \MT@ifempty{#2}\let\@tempc\@empty{\MT@preset@aux@factor{#2}\@tempc}%
1429 \MT@ifempty{#3}\let\@tempb\@empty{\MT@preset@aux@factor{#3}\@tempb}%
1430 \else
1431 \MT@ifempty{#1}\let\@tempa\@empty{\MT@preset@aux@space2{#1}\@tempa}%
1432 \MT@ifempty{#2}\let\@tempc\@empty{\MT@preset@aux@space3{#2}\@tempc}%
1433 \MT@ifempty{#3}\let\@tempb\@empty{\MT@preset@aux@space4{#3}\@tempb}%
1434 \fi

```

```

1435 \MT@set@all@sp\@tempa\@tempc\@tempb
1436 }
1437 }\relax

```

#### 14.2.4 Additional kerning

`\MT@kerning` Again, only check for additional kerning for new versions of pdf<sub>T</sub><sub>E</sub>X.

```

1438 \MT@requires@pdftex6{
1439 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

1440 \def\MT@set@kn@codes{%
1441 \MT@if@list@exists{%
1442 \MT@get@font@dimen@six
1443 \MT@get@opt
1444 \MT@reset@kn@codes
1445 \MT@get@inh@list
1446 \MT@set@inputenc{c}%
1447 \MT@load@list\MT@kn@cc@name
1448 \MT@set@listname
1449 \MT@let@cn\@tempc{MT@kn@cc@MT@kn@cc@name}%
1450 \expandafter\MT@set@codes\@tempc,\relax,%
1451 }\MT@reset@kn@codes
1452 }

```

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

```

1453 \def\MT@kn@split@val#1,#2\relax{%
1454 \def\@tempb{#1}%
1455 \MT@ifempty\@tempb\relax{%
1456 \MT@get@space@unit2%
1457 \MT@scale@to@em
1458 \knbcode\MT@font\MT@char=\@tempcntb
1459 (debug)\MT@info@n1{4}{;;; knbc (\MT@char): \number\knbcode\MT@font\MT@char: [#1]}%
1460 }%
1461 \def\@tempb{#2}%
1462 \MT@ifempty\@tempb\relax{%
1463 \MT@get@space@unit2%
1464 \MT@scale@to@em
1465 \knacode\MT@font\MT@char=\@tempcntb
1466 (debug)\MT@info@n1{4}{;;; knac (\MT@char): \number\knacode\MT@font\MT@char: [#2]}%
1467 }%
1468 \MT@ifdefined@c@T\MT@kn@inh@name{%
1469 \MT@ifdefined@n@T{MT@inh@MT@kn@inh@name @\MT@char @}{%
1470 \MT@exp@cs\MT@map@tlist@c{MT@inh@MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1471 }%
1472 }%
1473 }

```

`\MT@set@kn@heirs`

```

1474 \def\MT@set@kn@heirs#1{%
1475 \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1476 \knacode\MT@font#1=\knacode\MT@font\MT@char
1477 (debug)\MT@info@n1{2}{-- heir of \MT@char: #1}%
1478 (debug)\MT@info@n1{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1479 (debug) \number\knacode\MT@font\MT@char}%
1480 }

```

`\MT@set@all@kn`

```

\MT@reset@kn@codes 1481 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1482 (debug)\MT@info@n1{3}{-- knac/knbc: setting all to #1/#2}%
1483 \let\MT@temp\@empty

```

```

1484 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbcode\MT@font\@tempcnta=#1\relax}}%
1485 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knaccode\MT@font\@tempcnta=#2\relax}}%
1486 \MT@do@font\MT@temp
1487 }
1488 \def\MT@reset@kn@codes@{\MT@set@all@kn@z@z@}
1489 \let\MT@reset@kn@codes\relax

\MT@preset@kn
\MT@preset@kn@ 1490 \def\MT@preset@kn{%
1491   \expandafter\expandafter\expandafter\MT@preset@kn@
1492   \csname MT@kn@cc@MT@kn@cc@name @preset\endcsname\@nil
1493 }
1494 \def\MT@preset@kn@#1,#2\@nil{%
1495   \ifx\MT@kn@unit@\empty
1496     \MT@warn@preset@towidth{kn}%
1497     \let\MT@preset@aux\MT@preset@aux@factor
1498   \else
1499     \def\MT@preset@aux{\MT@preset@aux@space2}%
1500   \fi
1501   \MT@ifempty{#1}{\let\@tempa\empty}{\MT@preset@aux{#1}\@tempa}%
1502   \MT@ifempty{#2}{\let\@tempb\empty}{\MT@preset@aux{#2}\@tempb}%
1503   \MT@set@all@kn\@tempa\@tempb
1504 }
1505 }\relax

```

### 14.2.5 Tracking

This only works with pdfTeX 1.40.

```

1506 \MT@requires@pdftex6{

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

\MT@tr@font@list 1507 \let\MT@tr@font@list\empty
1508 \def\MT@tracking@{%
1509   \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
1510   \ifMT@inlist\else
1511     \MT@maybe@do{tr}%
1512     \ifMT@do\else
1513       \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,%}
1514     \fi
1515   \fi
1516 }
1517 </package>
1518 \let\MT@tracking
1519 <package> \MT@tracking@
1520 <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.

1521 \def\MT@set@tr@codes{%
1522   <*package>
1523   \MT@vinfo{Tracking font ~\MT@font'\on@line}%
1524   \MT@get@font@dimen@six
1525   \MT@if@list@exists
1526     \MT@get@tr@opt
1527     \relax
1528   </package>
1529   \MT@ifdefined@c@TF\MT@letterspace@\relax{\let\MT@letterspace@\MT@letterspace}%
1530   \ifnum\MT@letterspace@=\z@\else
1531   <package>    \MT@vinfo{... Tracking by \number\MT@letterspace@}%

```

Letterspacing only works in PDF mode.

1532 \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.

```
1533 \xdef\MT@lsfont{\csname\expandafter\string\font@name
1534 \number\MT@letterspace@ls\endcsname}%
1535 \expandafter\ifx\MT@lsfont\relax
1536 <debug>\MT@info@n1{1}{... new letterspacing instance}%
```

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

```
1537 \MT@get@ls@basefont
1538 \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@
```

Scale interword spacing (not configurable in letterspace).

```
1539 <*package>
1540 \MT@ifdefined@c@TF\MT@tr@ispace
1541 {\let\@tempa\MT@tr@ispace}%
1542 {\edef\@tempa{\MT@letterspace@*,,}}%
1543 \MT@ifdefined@c@TF\MT@tr@ospace
1544 {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1545 {\edef\@tempa{\@tempa,,}}%
1546 \expandafter\MT@tr@set@space\@tempa,%
1547 </package>
1548 <*letterspace>
1549 % spacing = {<letterspace amount>*,,}
1550 \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@relax sp
1551 * \fontdimen2\MT@lsfont/1000relax
1552 </letterspace>
```

Adjust outer kerning (microtype only).

```
1553 <*package>
1554 \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
1555 \expandafter\MT@tr@set@okern\@tempa,%
```

Disable ligatures (not configurable in letterspace).

```
1556 \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1557 </package>
1558 <*letterspace>
1559 % no ligatures = {f}
1560 \tagcode\MT@lsfont`f=\m@ne
1561 </letterspace>
```

Adjust protrusion values now, and maybe later (in `\MT@pr@split@val`).

```
1562 <debug>\MT@info@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%
1563 \MT@do@font{\lpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2relax
1564 \rprcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2relax}%
1565 <package> \let\MT@the@pr@code\MT@the@pr@code@tr
1566 \fi
```

Finally, let the letterspaced font propagate.

```
1567 \aftergroup\MT@set@lsfont
1568 <package> \let\MT@font\MT@lsfont
```

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

```
\MT@curr@ls 1569 \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
1570 \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.

```
1571 <package>
1572 \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
1573 \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1574 \aftergroup\aftergroup\aftergroup\MT@set@curr@os
1575 \MT@tr@outer@
1576 </package>
```

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

```
1577 \ifx\MT@ls@adjust\empty
1578 <letterspace> % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}
1579 \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1580 \MT@ls@outer@k
1581 <letterspace>
1582 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1583 \aftergroup\aftergroup\aftergroup\MT@set@curr@ok
1584 \aftergroup\aftergroup\aftergroup\MT@ls@outer@k
1585 </letterspace>
```

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

```
1586 <package>
1587 \else
1588 \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
1589 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1590 \ifdim\MT@outer@kern=\z@ \else \MT@ls@outer@k \fi
1591 \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
1592 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1593 </package>
1594 \fi
1595 <package>
```

`\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).

```
1596 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1597 \aftergroup\aftergroup\aftergroup\MT@set@curr@ok
1598 \aftergroup\aftergroup\aftergroup\MT@tr@outer@
1599 </package>
1600 \fi
1601 }
```

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

```
1602 <package>
1603 \def\MT@get@tr@opt{%
1604 \MT@set@listname
1605 \MT@ifdefined@n@T{MT@tr@c@MT@tr@c@name}{%
1606 \MT@let@cn\MT@letterspace{MT@tr@c@MT@tr@c@name}%
```

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

```
1607 \MT@ifdefined@n@T{MT@tr@c@MT@tr@c@name @unit}{%
1608 \MT@let@cn\MT@tr@unit@{MT@tr@c@MT@tr@c@name @unit}%
1609 \ifdim\MT@tr@unit@=1em
1610 \let\MT@tr@unit@\undefined
1611 \else
1612 \MT@let@cn\@tempb{MT@tr@c@MT@tr@c@name}%
1613 \MT@get@unit\MT@tr@unit@
1614 \let\MT@tr@factor@\@m
```

```

1615         \MT@scale@to@em
1616         \edef\MT@letterspace{\number\@tempcntb}%
1617     \fi
1618 }%
1619 }%

\MT@tr@ispace    Adjust interword spacing.
\MT@tr@ospace 1620 \MT@get@tr@opt@{spacing}    {ispace}%
1621 \MT@get@tr@opt@{outerspace}{ospace}%

\MT@tr@okern    Adjust outer kerning.
1622 \MT@get@tr@opt@{outerkerning}{okern}%

\MT@tr@ligatures    Which ligatures should we disable (empty means all, undefined none)?
1623 \MT@get@tr@opt@{noligatures} {ligatures}%
1624 }

\MT@get@tr@opt@
1625 \def\MT@get@tr@opt@#1#2{%
1626 \MT@ifdefined@n@T{MT@tr@c@{MT@tr@c@name @#1}}%
1627 { \MT@let@nn{MT@tr@#2}{MT@tr@c@{MT@tr@c@name @#1}}}%
1628 }
1629 /package

\MT@set@lsfont    Redefine \font@name, which will be called a second later (in \selectfont).
1630 plain \MT@requires@latex2{
1631 \def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}

\lsstyle    Disable the tests whether the font should be letterspaced, then trigger the setup.
1632 \DeclareRobustCommand\lsstyle{%
1633 package \def\MT@feat{tr}%
1634 \let\MT@tracking\MT@set@tr@codes
1635 \selectfont
1636 }

    Now the definitions for the letterspace package with plain TEX.
1637 plain
1638 }{
1639 \def\MT@set@lsfont{\MT@lsfont}
1640 \def\lsstyle{%
1641 \begingroup
1642 \escapechar\m@ne
1643 \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1644 \MT@set@tr@codes
1645 \endgroup
1646 }
1647 \let\textls\@undefined
1648 \let\lslig\@undefined
1649 }
1650 /plain

\lslig    For Fraktur fonts, some ligatures shouldn't be broken up. This command will
\MT@lslig    temporarily select the base font and insert the correct kerning.
1651 \DeclareRobustCommand\lslig[1]{%
1652 {\MT@ifdefined@c@TF\MT@curr@ls{%
1653 \escapechar\m@ne
1654 \MT@get@ls@basefont
1655 \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1656 \kern\MT@outer@kern
1657 \font@name #1%
1658 \kern\MT@outer@kern%
1659 }{#1}}}%

```



```

1660 }

\MT@get@ls@basefont pdfTeX 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, could destroy all
    previously set up micro-typographic features of the font.
1661 \def\MT@get@ls@basefont{%
1662   \edef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1663   \expandafter\ifx\MT@ls@basefont\relax
1664     \MT@exp@two@c\MT@glet\MT@ls@basefont\font@name
1665   \else
1666     (debug)\MT@info@nl{1}{... fixing base font}%
1667     \MT@exp@two@c\let\font@name\MT@ls@basefont
1668   \fi
1669 }

\MT@tr@noligatures pdfTeX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.
1670 (*package)
1671 \MT@requires@pdftex7{
1672   \def\MT@tr@noligatures{%
1673     \ifx\MT@tr@ligatures\empty
1674       \MT@noligatures@\MT@lsfont\undefined
1675     \else
1676       \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1677     \fi
1678   }
1679 }{
1680   \def\MT@tr@noligatures{%
1681     \MT@warning@nl{%
1682       Disabling selected ligatures is only possible since\MessageBreak
1683       pdftex 1.40.4. Disabling all ligatures instead}%
1684     \MT@glet\MT@tr@noligatures\relax
1685   }
1686 }

\MT@outer@space A new skip for outer spacing.
1687 \newskip\MT@outer@space

\MT@tr@set@space Adjust interword spacing (\fontdimen 2–4) 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.
1688 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
1689   (debug)\MT@info@nl2{... orig. space: \the\fontdimen2\MT@lsfont,
1690   (debug) \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
1691   (debug) \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
1692   \let\MT@temp\empty
1693   \MT@tr@set@space@{#1}{#4}{2}\empty
1694   \MT@tr@set@space@{#2}{#5}{3}\plus
1695   \MT@tr@set@space@{#3}{#6}{4}\minus
1696   \MT@glet@c\MT@outer@space\expandafter\string\font@name\MT@temp
1697   (debug)\MT@info@nl2{... inner space: \the\fontdimen2\MT@lsfont,
1698   (debug) \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
1699   (debug)\MT@info@nl2{... outer space: \MT@temp}%
1700 }

\MT@tr@set@space@ If outer spacing settings don't exist, they will be inherited from the inner spacing
settings.
1701 \def\MT@tr@set@space@#1#2#3#4{%
1702   \MT@ifempty{#2}{%

```

```

1703 \MT@ifempty{#1}{%
1704 \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
1705 }{%
1706 \MT@tr@set@space@@{#1}{#3}{1000}%
1707 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1708 \fontdimen#3\MT@lsfont=\@tempdima
1709 }%
1710 }{%
1711 \MT@tr@set@space@@{#2}{#3}{2000}%
1712 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1713 \MT@ifempty{#1}\relax{%
1714 \MT@tr@set@space@@{#1}{#3}{1000}%
1715 \fontdimen#3\MT@lsfont=\@tempdima
1716 }%
1717 }%
1718 }

```

`\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.

```

1719 \def\MT@tr@set@space@@#1#2#3{%
1720 \MT@test@ast#1*\@nil{%
1721 \MT@ifdefined@cTF\MT@tr@unit@
1722 {\edef\@tempb{#1}\MT@scale@to@em}
1723 {\@tempcntb=#1\relax}%
1724 \@tempdima=\dimexpr \dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
1725 -\fontdimen#2\MT@lsfont\relax

```

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

```

1726 \ifnum#2=\tw@
1727 \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
1728 \fi
1729 \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
1730 }{%
1731 \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
1732 \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
1733 }%
1734 (debug)\MT@info@n13{... : font dimen #2 (#1): \the\@tempdima}%
1735 }

```

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

```

1736 \def\MT@tr@outer@l{%
1737 \ifhmode
1738 \ifdim\lastskip>5sp
1739 \edef\x{\the\lastskip minus 0pt}%
1740 \setbox\z@\hbox{\MT@outer@space=\x}%
1741 \ifdim\wd\z@>\z@
1742 (debug)\MT@info@2{[[[ adjusting pre space: \the\MT@outer@space}%
1743 \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

1744 \let\MT@ls@outer@k\relax
1745 \else

```

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

```

1746 \ifdim\lastskip=%
1747 \ifnum\spacefactor<2000
1748 \spaceskip
1749 \else
1750 \ifdim\xspaceskip=\z@
1751 \dimexpr\spaceskip+\fontdimen7\font@name\relax

```

```

1752         \else
1753         \xspaceskip
1754         \fi
1755     \fi
1756 (debug) \MT@info2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
1757           \unskip \hskip\MT@outer@space\relax
1758           \let\MT@ls@outer@k\relax
1759         \fi
1760     \fi
1761 \fi
1762 \fi
1763 }

```

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

```

\MT@tr@outer@r@ 1764 \def\MT@tr@outer@r@{%
1765   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
1766 }
1767 \def\MT@tr@outer@r@{%
1768   \def\MT@temp*##1{\ifhmode\hskip\MT@outer@space
1769 (debug) \MT@info2{[[[ adjusting post spaces (1): \the\MT@outer@space}%
1770     \fi}%
1771   \ifcat\egroup\noexpand\MT@tr@outer@next
1772     \ifhmode\unkern\fi\egroup
1773     \MT@set@curr@ok
1774     \MT@set@curr@os
1775     \def\MT@temp*{\afterassignment\MT@tr@outer@r@\let\x=}%
1776   \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.

```

1777   \ifx\maybe@ic\MT@tr@outer@next
1778     \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\x=}%
1779   \else

```

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).

```

1780   \ifx\check@icr\MT@tr@outer@next
1781     \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\x=}%
1782   \else
1783     \ifx\@sptoken\MT@tr@outer@next
1784       \def\MT@temp* {\ifhmode\hskip\MT@outer@space
1785 (debug) \MT@info2{[[[ adjusting post spaces (2): \the\MT@outer@space}%
1786       \fi}%
1787     \else
1788       \ifx-\MT@tr@outer@next
1789         \def\MT@temp*~{\nobreak\hskip\MT@outer@space
1790 (debug) \MT@info2{[[[ adjusting post spaces (3): \the\MT@outer@space}%
1791         }%
1792     \else
1793       \ifx\ \MT@tr@outer@next \else
1794       \ifx\space\MT@tr@outer@next \else
1795       \ifx\@xobeysp\MT@tr@outer@next \else

```

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

```

1796       \def\MT@temp*{\ifdim\MT@outer@kern=\z@ \else\MT@ls@outer@k\fi}%
1797       \let\MT@tr@outer@next\relax

```

```

1798 \fi\fi\fi\fi\fi\fi\fi\fi
1799 \MT@temp*%
1800 }

```

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

```

\MT@tr@outer@icr@ 1801 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
1802 \def\MT@tr@outer@icr@{%
1803 \let\@let@token= \MT@tr@outer@next
1804 \maybe@ic@
1805 }

```

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

```

1806 }{
1807 \DeclareRobustCommand\lsstyle{%
1808 \MT@error{Letterspacing only works with pdfTeX version 1.40\MessageBreak
1809 or newer}{Upgrade pdfTeX, or use the 'soul' package instead.}%
1810 \MT@glet\lsstyle\relax
1811 }
1812 }
1813 </package>

```

`\textls` This command may be used like the other text commands.

```

1814 \DeclareRobustCommand\textls{%
1815 \hmode\bgroup
1816 \@ifstar{\let\MT@ls@adjust\@empty\MT@textls}%
1817 {\let\MT@ls@adjust\relax \MT@textls}%
1818 }

```

`\MT@textls` The optional argument may be used to change the letterspacing factor.

```

\MT@letterspace@ 1819 \newcommand\MT@textls[2][{}]{%
1820 \MT@ifempty{#1}%
1821 {\let\MT@letterspace@\@undefined}%
1822 {\KV@esp@def\MT@letterspace@{#1}%
1823 \MT@ls@too@large\MT@letterspace@}%
1824 \lsstyle #2%
1825 \egroup
1826 }

```

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

```

1827 \def\MT@ls@too@large#1{%
1828 \ifnum#1>\MT@tr@max
1829 \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
1830 \let#1\MT@tr@max
1831 \else
1832 \ifnum#1<\MT@tr@min
1833 \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
1834 \let#1\MT@tr@min
1835 \fi
1836 \fi
1837 }

```

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

```

\MT@tr@set@okern 1838 \newdimen\MT@outer@kern
1839 <*package>
1840 \def\MT@tr@set@okern#1,#2,{%
1841 \let\MT@temp\@empty
1842 \MT@tr@set@okern@{#1}%
1843 \MT@tr@set@okern@{#2}%
1844 \MT@glet@cnc\MT@outer@kern\expandafter\string\font@name}\MT@temp
1845 <debug>\MT@dinfo@n12{... outer kerning: (#1,#2)
1846 <debug> = \@nameuse{\MT@outer@kern\expandafter\string\font@name}}%

```

```

1847 }

\MT@tr@set@okern@

1848 \def\MT@tr@set@okern@#1{%
1849   \MT@test@ast#1*\@nil{%
1850     \MT@ifdefined@c@TF\MT@tr@unit@
1851     {\edef\@tempb{#1}\MT@scale@to@em}
1852     {\@tempcntb=#1\relax}%
1853     \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
1854   }{%
1855     \MT@ifempty\@tempa{\let\@tempa\@m}\relax
1856     \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
1857     * \fontdimen6\MT@lsfont/2000\relax
1858   }%
1859   \advance\@tempdima -\dimexpr \MT@letterspace@ sp
1860   * \fontdimen6\MT@lsfont/2000\relax
1861   \edef\MT@temp{\MT@temp{\the\@tempdima}}%
1862 }
1863 </package>

\MT@ls@outer@k    Adjust outer kerning.

1864 \def\MT@ls@outer@k{\ifhmode\kern\MT@outer@kern\relax\fi}
1865 <*package>

```

### 14.2.6 Disabling ligatures

\MT@noligatures     The possibility to disable ligatures is a new features of pdf $\TeX$  1.30.

```

1866 \MT@requires@pdftex5{
1867 \def\MT@noligatures{%
1868   \MT@dotrue
1869   \let\@tempa\MT@nl@setname
1870   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1871     \MT@ifdefined@c@TF\MT@checklist@##1}%
1872     {\csname MT@checklist@##1\endcsname}%
1873     {\MT@checklist@##1}}%
1874   {nl}%
1875 }%
1876 \ifMT@do
1877   \MT@noligatures@\MT@font\MT@nl@ligatures
1878 \fi
1879 }

```

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

```

1880 \def\MT@noligatures@#1#2{%
1881   \MT@ifdefined@c@TF#2{%

```

Early Mi $\TeX$  versions (before 2.5.2579) didn't know \tagcode.

```

1882   \MT@ifdefined@c@TF\tagcode{%

```

No 'inputenc' key.

```

1883     \let\MT@warn@maybe@inputenc\@empty
1884     \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
1885     \MT@map@clist@c#2{%
1886       \KV@sp@def\@tempa{##1}\MT@get@slot
1887       \ifnum\MT@char>\m@ne \tagcode#1\MT@char=\m@ne \fi}%
1888     \MT@vinfo{... Disabling ligatures for characters: #2}%
1889   }{%
1890     \pdfnoligatures#1%
1891     \MT@warning{Cannot disable selected ligatures (pdfTeX doesn't\MessageBreak
1892       know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
1893       the font instead}%

```

```

1894 }%
1895 }{%
1896 \pdfnoligatures#1%
1897 \MT@vinfo{... Disabling ligatures}%
1898 }%
1899 }
1900 }\relax

```

## 14.2.7 Loading the configuration

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

```

1901 \def\MT@load@list#1{%
1902 \edef\@tempa{#1}%
1903 \MT@let@cn\@tempb{\MT@MT@feat @c@\@tempa @load}%
1904 \MT@ifstreq\@tempa\@tempb{%
1905 \MT@error{\@nameuse{\MT@abbr@\MT@feat} list `@\@tempa' cannot load itself}{}%
1906 }{%
1907 \ifx\@tempb\relax \else
1908 \MT@ifdefined@n@TF{\MT@\MT@feat @c@\@tempb}{%
1909 \MT@vinfo{... : First loading \@nameuse{\MT@abbr@\MT@feat} list `@\@tempb'}%
1910 \begingroup
1911 \MT@load@list\@tempb
1912 \endgroup
1913 \edef\MT@curr@list@name{\@nameuse{\MT@abbr@\MT@feat} list
1914 \noexpand\MessageBreak`@\@tempb'}%
1915 \MT@let@cn\@tempc{\MT@\MT@feat @c@\@tempb}%
1916 \expandafter\MT@set@codes\@tempc,\relax,%
1917 }{%
1918 \MT@error{\@nameuse{\MT@abbr@\MT@feat} list `@\@tempb' undefined.\MessageBreak
1919 Cannot load it from list `@\@tempa'}{}%
1920 }%
1921 \fi
1922 }%
1923 }

```

`\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.

```

1924 \let\MT@file@list\empty
1925 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

1926 \MT@in@clist{#1}\MT@file@list
1927 \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.

```

1928 \MT@begin@catcodes
1929 \let\MT@begin@catcodes\relax
1930 \let\MT@end@catcodes\relax
1931 \InputIfFileExists{mt-#1.cfg}{%
1932 \edef\MT@curr@file{mt-#1.cfg}%
1933 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1934 \MT@xadd\MT@file@list{#1,%}
1935 }{%
1936 \MT@get@basefamily#1\empty\empty\empty\nil
1937 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1938 \ifMT@inlist@
1939 \MT@xadd\MT@file@list{#1,%}
1940 \else
1941 \InputIfFileExists{mt-\@tempa.cfg}{%

```

```

1942         \edef\MT@curr@file{mt-\@tempa.cfg}%
1943         \MT@vinfo{... Loading configuration file \MT@curr@file}%
1944         \MT@xadd\MT@file@list{\@tempa,#1,}%
1945     }{%
1946         \MT@vinfo{... No configuration file mt-#1.cfg}%
1947         \MT@xadd\MT@file@list{#1,}%
1948     }%
1949     \fi
1950 }%
1951 \endgroup
1952 \fi
1953 }

```

`\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  $\text{\LaTeX}$  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.

```

1954 \def\MT@cfg@catcodes{%
1955     \makeatletter
1956     \catcode`\^7%
1957     \catcode`\_9%
1958     \catcode`\^^I9%
1959     \catcode`\^^M9%
1960     \catcode`\z0
1961     \catcode`\{\@ne
1962     \catcode`\}\@tw@
1963     \catcode`\#6%
1964     \catcode`\%14%
1965     \MT@map@tlist@n
1966     {\!\"$&'\"(\)\*+,\-\.\/\:\;\<=\>\?[\]\_-\|/~}%
1967     \@makeother
1968 }

```

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

```

1969 \def\MT@begin@catcodes{%
1970     \begingroup
1971     \MT@cfg@catcodes
1972 }

```

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

```

1973 \let\MT@end@catcodes\endgroup

```

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

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

```

1974 \def\MT@get@basefamily#1#2#3#4\@nil{%
1975     \ifx\@empty#4%
1976         \def\@tempa{#1#2#3}%
1977     \else
1978         \let\@tempa\@empty
1979         \edef\@tempb{#1#2#3#4}%
1980         \expandafter\MT@get@basefamily@\@tempb\@nil
1981     \fi

```

Table 4: Order for matching font attributes

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

1982 }

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

```

1983 \def\MT@get@basefamily@#1#2\@nil{%
1984   \edef\@tempa{\@tempa#1}%
1985   \ifx\#2\@expandafter\@gobble\else\expandafter\@firstofone\fi
1986   {\MT@in@tlist{#2}\MT@variants
1987    \ifMT@inlist\else\MT@get@basefamily@#2\@nil\fi}%
1988 }
```

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

```

\MT@get@listname
\MT@get@listname@
\MT@get@listname@
1989 \def\MT@get@listname#1{%
1990   (debug)\MT@info{n1}{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
1991   \let\MT@listname\@undefined
1992   \def\@tempb{#1}%
1993   \MT@map@tlist@c\MT@try@order\MT@get@listname@
1994 }
1995 \def\MT@get@listname@#1{%
1996   \expandafter\MT@next@listname#1%
1997   \ifx\MT@listname\@undefined \else
1998     \expandafter\MT@tlist@break
1999   \fi
2000 }
```

\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.

```

2001 \def\MT@try@order{%
2002   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2003   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2004 }
```

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

```

2005 \def\MT@next@listname#1#2#3#4{%
2006   \edef\@tempa{\MT@encoding
2007     /\ifnum#1=\@ne \MT@family\fi
2008     /\ifnum#2=\@ne \MT@series\fi
2009     /\ifnum#3=\@ne \MT@shape\fi
2010     /\ifnum#4=\@ne *\fi
2011     \MT@context}%
2012   (debug)\MT@info{n1}{1}{trying \@tempa}%
2013   \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
2014     \MT@next@listname@#4%
2015   }{%
```

Also try with an alias family.

```

2016   \ifnum#1=\@ne
```



```

2017 \ifx\MT@familyalias\empty \else
2018 \edef\@tempa{\MT@encoding
2019 \MT@familyalias
2020 /\ifnum#2=\@ne \MT@series\fi
2021 /\ifnum#3=\@ne \MT@shape\fi
2022 /\ifnum#4=\@ne *\fi
2023 \MT@context}%
2024 (debug)\MT@edinfo{n1}{1}{(alias) \@tempa}%
2025 \MT@ifdefined@nT{\MT@\@tempb @\@tempa}{%
2026 \MT@next@listname@#4%
2027 }%
2028 \fi
2029 \fi
2030 }%
2031 }

```

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

```

2032 \def\MT@next@listname@#1{%
2033 \ifnum#1=\@ne
2034 \MT@exp@cs\MT@in@rlist{\MT@\@tempb @\@tempa @sizes}%
2035 \ifMT@inlist@
2036 \let\MT@listname\MT@size@name
2037 \fi
2038 \else
2039 \MT@let@cn\MT@listname{\MT@\@tempb @\@tempa}%
2040 \fi
2041 }

```

\MT@if@list@exists

```

\MT@context 2042 \def\MT@if@list@exists{%
2043 \MT@let@cn\MT@context{\MT@\MT@feat @context}%
2044 \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
2045 \MT@get@listname{\MT@feat @c}%
2046 \MT@ifdefined@c@TF\MT@listname{%
2047 \MT@edef@n{\MT@\MT@feat @c@name}{\MT@listname}%
2048 \ifMT@nonselected
2049 \MT@vinfo{... Applying non-selected expansion (list '\MT@listname')}%
2050 \else
2051 \MT@vinfo{... Loading \@nameuse{\MT@abbr@\MT@feat} list '\MT@listname'}%
2052 \fi
2053 \@firstoftwo
2054 }%

```

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

```

2055 \MT@let@nc{\MT@\MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

2056 \ifMT@nonselected
2057 \MT@vinfo{... Applying non-selected expansion (no list)}%
2058 \else

```

Tracking doesn't require a list, either.

```

2059 \MT@ifstreq\MT@feat{tr}\relax%
2060 \MT@warning{I cannot find a \@nameuse{\MT@abbr@\MT@feat} list
2061 for font\MessageBreak'\MT@@font'%
2062 \ifx\MT@context\@empty\else\space(context: '\MT@context')\fi.
2063 Switching off\MessageBreak\@nameuse{\MT@abbr@\MT@feat} for this font}%
2064 }%
2065 \fi
2066 \@secondoftwo
2067 }%

```

```

2068 }
\MT@get@inh@list    The inheritance lists are global (no context).
\MT@context 2069 \def\MT@get@inh@list{%
2070   \let\MT@context\@empty
2071   \MT@get@listname{\MT@feat @inh}%
2072   \MT@ifdefined@c@TF\MT@listname{%
2073     \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
2074     <debug>\MT@info@n{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
2075     <debug>           '\MT@listname'}%
2076     \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%
    If the list is \@empty, it has already been parsed.
2077     \ifx\@tempc\@empty \else
2078     <debug>\MT@info@n{1}{parsing inheritance list ...}%
    The group is only required in case an input encoding is given.
2079     \begingroup
2080     \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak'\MT@listname'}%
2081     \MT@set@inputenc{inh}%
2082     \expandafter\MT@inh@do\@tempc,\relax,%
2083     \MT@gl@et@nc{MT@\MT@feat @inh@\MT@listname}\@empty
2084     \endgroup
2085     \fi
2086   }%
2087   \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
2088 }%
2089 }

```

### 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@ 2090 \def\MT@get@slot{%
2091   \escapechar~\
2092   \let\MT@char@m@ne
2093   \MT@noresttrue

```

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

```

2094   \MT@toks=\expandafter{\@tempa}%

```

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

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

```

2095   \expandafter\MT@is@letter\@tempa\relax\relax
2096   \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.

```

2097   \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.

```
2098 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
2099 \MT@is@symbol
```

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

```
2100 {\expandafter\MT@is@composite\@tempa\relax\relax}%
2101 \ifnum\MT@char@ < \z@
```

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

```
2102 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2103 \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2104 \fi
2105 \fi

2106 \let\MT@char\MT@char@
2107 \ifnum\MT@char < \z@
2108 \MT@warn@unknown
2109 \else
```

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.

```
2110 \ifMT@norest \else
2111 \MT@warn@rest
2112 \let\MT@char\m@ne
2113 \fi
2114 \fi
2115 \escapechar\m@ne
2116 }
```

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

```
2117 \newif\ifMT@norest
```

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

```
2118 \def\MT@is@letter#1#2\relax{%
2119 \ifcat a\noexpand#1\relax
2120 \edef\MT@char@{\number`#1}%
2121 \ifx\#2\%
2122 (debug)\MT@info@n{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2123 \else
2124 \MT@norestfalse
2125 \fi
2126 \else
2127 \ifcat !\noexpand#1\relax
2128 \edef\MT@char@{\number`#1}%
2129 (debug)\MT@info@n{3}{> `the\MT@toks' is a character (\MT@char@)}%
2130 \ifx\#2\%
2131 \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
2132 \else
2133 \MT@norestfalse
2134 \expandafter\MT@is@number#1#2\relax\relax
2135 \fi
2136 \fi
2137 \fi
```

2138 }

`\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.

```

2139 \def\MT@is@number#1#2#3\relax{%
2140   \ifx\relax#3\relax \else
2141     \ifx\relax#2\relax \else
2142       \MT@noesttrue
2143       \if#1"\relax
2144         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2145   <debug>\MT@info@n1{3}{> ... a hexadecimal number: \MT@char@}%
2146     \else
2147       \if#1'\relax
2148         \def\MT@char@{\number#1#2#3}%
2149   <debug>\MT@info@n1{3}{> ... an octal number: \MT@char@}%
2150     \else
2151       \MT@ifint{#1#2#3}{%
2152         \def\MT@char@{\number#1#2#3}%
2153   <debug>\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
2154       }\MT@noestfalse
2155     \fi
2156   \fi
2157   \ifnum\MT@char@ > \@cclv
2158     \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2159     \let\MT@char@\m@ne
2160   \fi
2161 \fi
2162 \fi
2163 }
```

`\MT@is@active` Expand an active character. (This was completely broken in v1.7, and only worked by chance before.) We `\set@display@protect` to translate, e. g., Ä into `\ "A`, that is to whatever it is defined in the inputenc encoding file.

Unfortunately, the (older) inputenc definitions prefer the protected/generic variants (e. g., `\copyright` instead of `\textcopyright`), which our parser won't be able to understand. (I'm fed up now, so you have to complain if you really, really want to be able to write '©' instead of `\textcopyright`, thus rendering your configuration files unportable.)

Unicode characters (inputenc/utf8,utf8x) are also supported.

```

2164 \def\MT@is@active#1#2\@nil{%
2165   \ifnum\catcode`#1 = \active
2166     \begingroup
2167     \set@display@protect
2168     \let\IeC\@firstofone
2169     \let\@inpenc@undefined@\MT@undefined@char
```

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

```

2170   \def\UTFviii@defined##1{\ifx ##1\relax
2171     \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
```

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

```

2172   \MT@ifdefined@c@T\PrerenderUnicode
2173   {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
2174   \edef\x{\endgroup
2175     \def\noexpand\@tempa{\@tempa}%
```

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

```
2176 \MT@toks={\the\MT@toks\space(=\@tempa)}%
2177 }%
2178 \x
2179 \fi
2180 }
```

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

```
2181 \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.

```
2182 \def\MT@is@symbol{%
2183   \expandafter\def\expandafter\MT@char\expandafter
2184     {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2185   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2186     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2187   \ifnum\MT@char@ < \z@
```

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

```
2188   \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2189   \fi
2190 }
```

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

```
\MT@charstring 2191 \begingroup
2192   \catcode`\=/\z@
2193   /MT@map@tlist@n{/CHAR}/@makeother
2194   /lowercase{%
2195     /def/x{/endgroup
2196       /def/MT@charstring{CHAR"%
2197       /def/MT@is@char##1CHAR"##2##3##4/relax{%
2198         /ifx/relax##1/relax
2199         /if##3\relax
2200         /edef/MT@char@{/number"##2}%
2201         /MT@ifstreq/MT@charstring{##3##4}/relax/MT@noestfalse
2202         /else
2203         /edef/MT@char@{/number"##2##3}%
2204         /MT@ifstreq/MT@charstring{##4}/relax/MT@noestfalse
2205         /fi
2206         <debug> /MT@dinfo@n1{3}{> ~/the/MT@toks' is a \char (/MT@char@)}%
2207         /fi
2208       }%
2209     }%
2210   }
2211 /x
```

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

```
2212 \def\MT@is@composite#1#2\relax{%
2213   \ifx\#2\\\else
```

Again, we construct a control sequence, this time of the form: `\<encoding>\<accent>-<character>`, e. g., `\T1"-a`, which we then expand once to see if it is a letter (if it has been defined by `\DeclareTextComposite`). This should be robust, finally, especially, since we also `\detokenize` the input instead of only `\stringifying`

it. Thus, we will die gracefully even on wrong Unicode input without utf8.

```

2214 \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2215 \string\csname\MT@encoding\endcsname
2216 \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
2217 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2218 \fi
2219 }

```

[What about math? Well, for a moment the following looked like a solution, with `\mt@is@mathchar` defined accordingly, analogous to `\MT@is@char` above, to pick up the last two tokens (the `\meaning` of a `\mathchardef`'ed command expands to its hexadecimal notation):

```

\def\MT@is@mathchar#1{%
  \if\relax\noexpand#1% it's a macro
    \let\x#1%
  \else % it's a character
    \mathchardef\x=\mathcode`#1\relax
  \fi
  \expandafter\MT@exp@two@c\expandafter\mt@is@mathchar\expandafter
  \meaning\expandafter\x\mt@mathcharstring\relax\relax\relax
}

```

However, the problem is that `\mathcodes` and `\mathchardefs` have global scope. Therefore, if they are changed by a package that loads different math fonts, there is no guarantee whatsoever that things will still be correct (e. g., the minus in `cmsy` when the `euler` package is loaded). So, no way to go, unfortunately.]

Some warning messages, for performance reasons separated here.

`\MT@curr@list@name` The type and name of the current list, defined at various places.

```

\MT@set@listname 2220 \def\MT@set@listname{%
2221 \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
2222 ~\@nameuse{MT@\MT@feat @c@name}}}%
2223 }

```

`\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.

```

2224 \def\MT@warn@ascii{%
2225 \MT@warning@nl{Character '\the\MT@toks' (= \MT@char@)
2226 is outside of ASCII range.\MessageBreak
2227 You must load the 'inputenc' package before using\MessageBreak
2228 8-bit characters in \MT@curr@list@name}%
2229 }

```

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

```

2230 \def\MT@warn@number@too@large#1{%
2231 \MT@warning@nl{%
2232 Number #1 in encoding '\MT@encoding' too large!\MessageBreak
2233 Ignoring it in \MT@curr@list@name}%
2234 }

```

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

```

2235 \def\MT@warn@rest{%
2236 \MT@warning@nl{%
2237 Unknown slot number of character\MessageBreak'\the\MT@toks'%
2238 \MT@warn@maybe@inputenc\MessageBreak
2239 in font encoding '\MT@encoding'.\MessageBreak
2240 Make sure it's a single character\MessageBreak
2241 (or a number) in \MT@curr@list@name}%
2242 }

```

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

```
2243 \def\MT@warn@unknown{%
2244   \MT@warning@nl{%
2245     Unknown slot number of character\MessageBreak`the\MT@toks'%
2246     \MT@warn@maybe@inputenc\MessageBreak
2247     in font encoding `\'MT@encoding' in \MT@curr@list@name}%
2248 }
```

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

```
2249 \def\MT@warn@maybe@inputenc{%
2250   \MT@ifdefined@n@T
2251   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
2252   { (input encoding `\'@nameuse
2253   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2254 }
```

### 14.2.9 Hook into L<sup>A</sup>T<sub>E</sub>X's font selection

We append \MT@setupfont to \pickup@font, which is called by L<sup>A</sup>T<sub>E</sub>X every time a font is selected. We then check whether we've already seen this font, and if not, set it up for micro-typography. This ensures that we will catch all fonts, and that we will not set up fonts more than once. The whole package really hangs on this command.

In contrast to the pdfcpot package, it is not necessary to declare in advance which fonts should benefit from micro-typographic treatment. Also, only those fonts that are actually being used will be set up.

For my reference:

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

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

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

```
\MT@font 2255 \let\MT@font@list\@empty
2256 \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`.

```
2257 </package>
2258 <plain>\MT@requires@latex2{
2259 \MT@addto@setup{%
```

`\MT@orig@pickupfont` 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 expansion. (Tracking doesn't really work for other reasons.) Like us, CJK redefines `\pickup@font`.

```
2260 \ifpackageloaded{CJK}{
2261   \ifpackageafter{CJK}{2006/10/17} % 4.7.0
2262   {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}}
2263    {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}}
2264   \g@addto@macro\MT@orig@pickupfont
2265   {\expandafter\ifx\font@name\relax\define@newfont\fi}}
```

CJKut8 redefines `\pickup@font` once more (recent versions, in PDF mode, as determined by `ifpdf`, which CJKut8 loads).

```
2266 \ifpackageloaded{CJKut8}
2267 {\ifpackageafter{CJKut8}{2008/05/22} % 4.8.0
2268  {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}
2269  {\@firstoftwo}}
2270 {\@firstoftwo}
2271 {\g@addto@macro\MT@orig@pickupfont{
2272  {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2273   \define@newfont\else\xdef\font@name{
2274    \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}
2275 {\g@addto@macro\MT@orig@pickupfont{
2276  {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2277   \define@newfont\def\CJK@temp{v}%
2278   \ifx\CJK@temp\CJK@plane
2279    \expandafter\ifx\csname CJK@cmaph@f@family\CJK@plane\endcsname\relax
2280    \else\csname CJK@cmaph@f@family\CJK@plane\endcsname\fi
2281    \else \CJK@addcmaph\CJK@plane \fi
2282    \else\xdef\font@name{
2283     \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}
2284 }}
2285 {\def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
2286 }
```

Check whether `\pickup@font` is defined as expected. The warning issued by `\CheckCommand*` would be a bit too generic.

```
2287 \ifx\pickup@font\MT@orig@pickupfont \else
2288   \MT@warning@nl{%
2289     Command \string\pickup@font\space is not defined as expected.%
2290     \MessageBreak Patching it anyway. Some things may break%
2291   }<package>
2292   .\MessageBreak Double-check whether micro-typography is indeed%
2293   \MessageBreak applied to the document.%
2294   \MessageBreak (Hint: Turn on `verbose' mode)%
2295 </package>
2296 }
2297 \fi
```



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

```
2298 \g@addto@macro\pickup@font{\begingroup}
```

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

```
2299 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
2300 \g@addto@macro\pickup@font{%
2301   \escapechar\m@ne
2302 }*package>
2303 <debug> \global\MT@inannottrue
2304 <debug> \MT@glet\MT@pdf@annot\@empty
2305 <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.

```
2306 \MT@let@cn\MT@font{MT@subst@expandafter\string\font@name}%
2307 \ifx\MT@font\relax
2308   \let\MT@font\font@name
2309 \else
2310   \ifx\MT@font\font@name \else
2311 <debug> \MT@addto@annot{= substituted with \MT@@font}%
2312   \MT@register@subst@font
2313   \fi
2314 \fi
2315 \MT@setupfont
2316 }*package>
2317 <letterspace> \MT@tracking
2318 \endgroup
2319 }
2320 }*package>
```

`\MT@pickupfont` Remember the patched command for later.

```
2321 \let\MT@pickupfont\pickup@font
```

`\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.

```
2322 \g@addto@macro\do@subst@correction
2323 { \edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2324   \MT@glet@nc{MT@subst@expandafter\string\font@name}\MT@font}
```

`\add@accent` Inside `\add@accent`, we have to disable microtype's setup, since the grouping in `\MT@orig@add@accent` the patched `\pickup@font` would break the accent if different fonts are used for the base character and the accent. Fortunately,  $\text{\LaTeX}$  takes care that the fonts used for the `\accent` are already set up, so that we cannot be overlooking them.

```
2325 \let\MT@orig@add@accent\add@accent
2326 \def\add@accent#1#2{%
2327   \let\pickup@font\MT@orig@pickupfont
2328   \MT@orig@add@accent{#1}{#2}%
2329   \let\pickup@font\MT@pickupfont
2330 }
2331 }*package>
2332 }
2333 <plain>\relax
```

2334 *(\*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.

2335 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}

\MT@register@subst@font Register the substituted font.

2336 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}

\MT@register@font Register the current font.

2337 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}

#### 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.

2338 \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.

```
2339 \def\MT@check@font@cx{%
2340   \MT@if@true
2341   \MT@map@clist@c\MT@active@features{%
2342     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
2343     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
2344     \ifMT@inlist@
2345       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
2346     \else
2347       \MT@if@false
2348     \fi
2349   }%
2350   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2351 }
```

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

```
2352 \def\MT@register@subst@font@cx{%
2353   \MT@map@clist@c\MT@active@features{%
2354     \MT@exp@cs\MT@xadd
2355     {MT@##1\csname MT@##1@context\endcsname font@list}%
2356     {\font@name,}%
2357   }%
2358 }
```

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

```
2359 \def\MT@register@font@cx{%
2360   \MT@map@clist@c\MT@active@features{%
2361     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
2362       \MT@exp@cs\MT@xadd
2363       {MT@##1\csname MT@##1@context\endcsname font@list}%
2364       {\MT@font,}%
2365     \def\@tempa{##1}%
2366     \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
2367   \fi
2368   }%
2369 }
```

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

```
2370 \def\MT@maybe@rem@from@list#1{%
2371   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
2372     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
2373     \MT@font \csname MT@\@tempa @#1font@list\endcsname
2374   }%
2375 }
```

`\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.

```
2376 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}%
2377 \MT@addto@setup{%
2378   \DeclareRobustCommand\microtypecontext[1]{%
2379     \MT@setup@contexts
2380     \let\MT@reset@context\relax
2381     \setkeys{MTC}{#1}%
2382     \selectfont
2383     \MT@reset@context
2384   }%
2385 }
```

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

```
2386 \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@`

```
2387 \def\MT@reset@context@{%
2388   \MT@vinfo{<<< Resetting contexts\on@line
2389   \<debug> \MessageBreak= \MT@pr@context/\MT@ex@context
2390   \<debug> \MT@tr@context/\MT@kn@context/\MT@sp@context
2391   }%
2392   \selectfont
2393 }
```

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

```
2394 \def\MT@setup@contexts{%
2395   \MT@map@clist@c\MT@active@features
2396   {\MT@gl@et@c\MT@##1@font@list}\MT@font@list}%
2397   \MT@gl@et\MT@check@font\MT@check@font@cx
2398   \MT@gl@et\MT@register@font\MT@register@font@cx
2399   \MT@gl@et\MT@register@subst@font\MT@register@subst@font@cx
2400   \MT@gl@et\MT@setup@contexts\relax
2401 }
```

Define context keys.

```
2402 \MT@map@clist@c\MT@features@long{%
2403   \define@key{MTC}{#1}[]{}%
2404   \edef\@tempb{\@nameuse{MT@rbba#1}}%
2405   \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2406   \ifMT@inlist@
```

Using an empty context is only asking for trouble, therefore we choose the '0' instead (hoping for the L<sup>A</sup>T<sub>E</sub>X users' natural awe of this character).

```
2407   \MT@ifempty{#1}{\def\MT@val{0}}{\def\MT@val{#1}}%
2408   \MT@exp@cs\ifx\MT@\@tempb @context\MT@val
2409   \<debug>\MT@dinfo{1}{>>> no change of #1 context: '\MT@val'}%
2410   \else
```

```

2411 \MT@vinfo{>>> Changing #1 context to ` \MT@val'\MessageBreak\on@line
2412 (debug) \space(previous: ` \@nameuse{MT@\@tempb @context}')%
2413 }%
2414 \def\MT@reset@context{\aftergroup\MT@reset@context}%

```

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

```

2415 \MT@gl@et@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes}%

```

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

```

2416 \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
2417 \MT@val \csname MT@\@tempb @doc@contexts\endcsname
2418 \ifMT@inlist@ \else
2419 \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{\MT@val}%
2420 (debug) \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
2421 \fi
2422 \MT@edef@n{MT@\@tempb @context}{\MT@val}%
2423 \fi
2424 \fi
2425 }%
2426 }

```

\MT@pr@context Initialise the contexts.

```

\MT@ex@context 2427 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}%

```

```

\MT@tr@context 2428 \MT@def@n{MT@#1@context}{@}%

```

```

\MT@sp@context 2429 \MT@def@n{MT@#1@doc@contexts}{\@}%

```

```

2430 }

```

```

\MT@kn@context 2431 \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.

```

2432 \def\DeclareMicrotypeSet{%
2433 \ifstar
2434 \MT@DeclareSetAndUseIt
2435 \MT@DeclareSet
2436 }

```

```

\MT@DeclareSet

```

```

2437 \newcommand\MT@DeclareSet[3] [] {%
2438 \KV@sp@def\@tempa{#1}%
2439 \MT@ifempty\@tempa{%
2440 \MT@map@clist@c\MT@features{\MT@declare@sets{##1}{#2}{#3}}}%
2441 }%
2442 \MT@map@clist@c\@tempa{%
2443 \KV@sp@def\@tempa{##1}%
2444 \MT@ifempty\@tempa\relax{%
2445 \MT@is@feature{set declaration `#2'}{%
2446 \MT@exp@one@n\MT@declare@sets
2447 {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
2448 }%
2449 }%
2450 }}%

```

```

2451 }%
2452 }

\MT@DeclareSetAndUseIt
2453 \newcommand\MT@DeclareSetAndUseIt[3] [] {%
2454   \MT@DeclareSet[#1]{#2}{#3}%
2455   \UseMicrotypeSet[#1]{#2}%
2456 }

\MT@curr@set@name    We need to remember the name of the set currently being declared.
2457 \let\MT@curr@set@name\empty

\MT@declare@sets    Define the current set name and parse the keys.
2458 \def\MT@declare@sets#1#2#3{%
2459   \KV@sp@def\MT@curr@set@name{#2}%
2460   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
2461     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2462     \MT@gl@et@nc{MT@#1@list@size@\MT@curr@set@name}\empty
2463   }%
2464   \MT@gl@et@nc{MT@#1@set@@\MT@curr@set@name}\empty
2465   (debug)\MT@din@fo{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2466   \setkeys{MT@#1@set}{#3}%
2467 }

\MT@define@set@key@    <#1> = font axis, <#2> = feature.
2468 \def\MT@define@set@key@#1#2{%
2469   \define@key{MT@#2@set}{#1} [] {%
2470     \MT@gl@et@nc{MT@#2@list@#1@\MT@curr@set@name}\empty
2471     \MT@map@cl@ist@n{##1}{%
2472       \KV@sp@def\MT@val{###1}%
2473       \MT@get@highlevel{#1}%

We do not add the expanded value to the list ...
2474       \MT@exp@two@n@g@addto@macro
2475       {\csname MT@#2@list@#1@\MT@curr@set@name\expandafter\endcsname}%
2476       {\MT@val,}%
2477     }%

... but keep in mind that the list has to be expanded at the end of the preamble.
2478     \expandafter\g@addto@macro\expandafter\MT@font@sets
2479     \csname MT@#2@list@#1@\MT@curr@set@name\endcsname
2480     (debug)\MT@din@fo@n1{1}{-- #1: \@nameuse{MT@#2@list@#1@\MT@curr@set@name}}%
2481   }%
2482 }

\MT@get@highlevel    Saying, for instance, ‘family=rm*’ or ‘shape=bf*’ will expand to \rmdefault resp.
                     \bfdefault.
2483 \def\MT@get@highlevel#1{%
2484   \expandafter\MT@test@ast\MT@val*\@nil\relax{%

And ‘family = *’ will become \familydefault.
2485   \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
2486   \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.
2487 }%
2488 }

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

```

```

2489 \def\MT@test@ast#1*#2\@nil{%
2490   \def\@tempa{#1}%
2491   \MT@ifempty{#2}%
2492 }

```

\MT@font@sets Fully expand the font specification and fix catcodes for all font sets.

```

\MT@fix@font@set 2493 \let\MT@font@sets\empty
2494 \def\MT@fix@font@set#1{%
2495   \xdef#1{#1}%
2496   \global\@onelevel@sanitize#1%
2497 }

```

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

```

2498 \def\MT@define@set@key@size#1{%
2499   \define@key{MT@#1@set}{size}[]{}%
2500   \MT@map@clist@n{##1}{%
2501     \KV@esp@def\MT@val{###1}%
2502     \expandafter\MT@get@range\MT@val--\@nil
2503     \ifx\MT@val\relax \else
2504       \MT@exp@cs\MT@xadd
2505       {MT@#1list@size@MT@curr@set@name}%
2506       {{{\MT@lower}{\MT@upper}\relax}}%
2507     \fi
2508   }%
2509   <debug>\MT@info@n1{1}{-- size: \@nameuse{MT@#1list@size@MT@curr@set@name}}%
2510   }%
2511 }

```

Font sizes may also be specified as ranges. This has been requested by Andreas Böhmann, who has also offered valuable help in implementing this. Now, it is for instance possible to set up different lists for fonts with optical sizes. (The MinionPro project 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  $\{\langle lower\ bound \rangle\} \{\langle upper\ bound \rangle\} \{\langle list\ name \rangle\}$ .

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

```

\MT@lower 2512 \def\MT@get@range#1-#2-#3\@nil{%
2513   \MT@ifempty{#1}{%
2514     \MT@ifempty{#2}{%
2515       \let\MT@val\relax
2516     }%
2517     \def\MT@lower{0}%
2518     \def\MT@val{#2}%
2519     \MT@get@size
2520     \edef\MT@upper{\MT@val}%
2521   }%
2522   }%
2523   \def\MT@val{#1}%
2524   \MT@get@size
2525   \ifx\MT@val\relax \else
2526     \edef\MT@lower{\MT@val}%
2527     \MT@ifempty{#2}{%
2528       \MT@ifempty{#3}%
2529       {\def\MT@upper{-1}}%

```

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

```

2530   {\def\MT@upper{2048}}%
2531   }%
2532   \def\MT@val{#2}%
2533   \MT@get@size
2534   \ifx\MT@val\relax \else

```

```

2535 \MT@ifdim\MT@lower>\MT@val{%
2536 \MT@error{%
2537 Invalid size range (\MT@lower\space > \MT@val) in font set
2538 ~\MT@curr@set@name'.\MessageBreak Swapping sizes}}{%
2539 \edef\MT@upper{\MT@lower}%
2540 \edef\MT@lower{\MT@val}%
2541 }{%
2542 \edef\MT@upper{\MT@val}%
2543 }%
2544 \MT@ifdim\MT@lower=\MT@upper
2545 {\def\MT@upper{-1}}%
2546 \relax
2547 \fi
2548 }%
2549 \fi
2550 }%
2551 }

```

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

```

2552 \def\MT@get@size{%
    A single star would mean \sizedefault, which doesn't exist, so we define it to be
    \normalsize.
2553 \if*\MT@val\relax
2554 \def\@tempa{\normalsize}%
2555 \else
2556 \MT@let@cn\@tempa{\MT@val}%
2557 \fi
2558 \ifx\@tempa\relax \else

```

The relsize solution of parsing \@setfontsize does not work with the AMS classes, among others. I hope my hijacking doesn't do any harm. We redefine \set@fontsize, and not @setfontsize because some classes might define the size selection commands by simply using \fontsize (e. g., the a0poster class).

```

2559 \begingroup
2560 \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
2561 \@tempa\@nil
2562 \fi

```

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

```

2563 \MT@ifdimen\MT@val{%
2564 \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2565 \edef\MT@val{\strip@pt\@tempdima}%
2566 }{%
2567 \MT@warning{Could not parse font size ~\MT@val'\MessageBreak
2568 in font set ~\MT@curr@set@name'}%
2569 \let\MT@val\relax
2570 }%
2571 }

```

\MT@define@set@key@font

```

2572 \def\MT@define@set@key@font#1{%
2573 \define@key{MT@#1@set}{font}[]{%
2574 \MT@gl@et@nc{MT@#1@list@font@\MT@curr@set@name}\@empty
2575 \MT@map@clist@n{##1}{%
2576 \KV@esp@def\MT@val{###1}%
2577 \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*}}\relax
2578 \expandafter\MT@get@font\MT@val////\@nil
2579 \MT@exp@two@n@g@addto@macro
2580 {\csname MT@#1@list@font@\MT@curr@set@name\expandafter\endcsname}%

```

```

2581     {\MT@val,}%
2582   }%
2583   \expandafter\g@addto@macro\expandafter\MT@font@sets
2584     \csname MT@#1list@font@MT@curr@set@name\endcsname
2585   <debug>\MT@edinfo{n1}{1}{-- font: \nameuse{MT@#1list@font@MT@curr@set@name}}%
2586   }%
2587 }

```

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

```

2588 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2589   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
2590   \ifx\MT@val\relax\def\MT@val{0}\fi
2591   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2592   \let\MT@val\@tempb
2593 }

```

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

```

2594 \def\MT@get@font@#1#2#3#4#5#6{%
2595   \let\@tempb\@empty
2596   \def\MT@temp{#1/#2/#3/#4/#5}%
2597   \MT@get@axis{encoding}{#1}%
2598   \MT@get@axis{family}{#2}%
2599   \MT@get@axis{series}{#3}%
2600   \MT@get@axis{shape}{#4}%
2601   \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
2602   \MT@ifempty{#5}{%
2603     \MT@warn@axis@empty{size}{\string\normalsize}%
2604     \def\MT@val{*}%
2605   }{%
2606     \def\MT@val{#5}%
2607   }%
2608   \MT@get@size
2609 }

```

**\MT@get@axis**

```

2610 \def\MT@get@axis#1#2{%
2611   \def\MT@val{#2}%
2612   \MT@get@highlevel{#1}%
2613   \MT@ifempty\MT@val{%
2614     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2615     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2616   }\relax
2617   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2618 }

```

**\MT@warn@axis@empty**

```

2619 \def\MT@warn@axis@empty#1#2{%
2620   \MT@warning{#1 axis is empty in font specification\MessageBreak
2621     ~\MT@temp'. Using ~#2' instead}%
2622 }

```

We have finally assembled all pieces to define **\DeclareMicrotypeSet**'s keys. It is also used for **\DisableLigatures**.

```

2623 \MT@exp@one@n\MT@map@clist@n{\MT@features,n1}{%
2624   \MT@define@set@key@{encoding}{#1}%
2625   \MT@define@set@key@{family}{#1}%
2626   \MT@define@set@key@{series}{#1}%
2627   \MT@define@set@key@{shape}{#1}%
2628   \MT@define@set@key@size{#1}%
2629   \MT@define@set@key@font{#1}%
2630 }

```



`\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.

```

2631 \renewcommand*\UseMicrotypeSet[2] [] {%
2632   \KV@sp@def\@tempa{#1}%
2633   \MT@ifempty\@tempa{%
2634     \MT@map@clist@MT@features{{\MT@use@set{##1}{#2}}}%
2635   }%
2636   \MT@map@clist@MT@tempa{%
2637     \KV@sp@def\@tempa{##1}%
2638     \MT@ifempty\@tempa\relax{%
2639       \MT@is@feature{activation of set `#2'}{%
2640         \MT@exp@one@MT@use@set
2641         {\csname MT@rbba@\@tempa\endcsname}{#2}%
2642       }%
2643     }%
2644   }}%
2645 }%
2646 }

```

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

```

\MT@ex@setname 2647 \def\MT@use@set#1#2{%
\MT@tr@setname 2648   \KV@sp@def\@tempa{#2}%
2649   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@sp@setname 2650     \MT@xdef@n{MT@#1@setname}{\@tempa}%
\MT@kn@setname 2651   }%
2652   \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
\MT@use@set 2653     \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2654   }%
2655   \MT@error{%
2656     The \@nameuse{MT@abbr@#1} set `@\@tempa' is undeclared.\MessageBreak
2657     Using set `@\@nameuse{MT@#1@setname}' instead}}%
2658   }%
2659 }

```

`\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.

```

2660 \renewcommand*\DeclareMicrotypeSetDefault[2] [] {%
2661   \KV@sp@def\@tempa{#1}%
2662   \MT@ifempty\@tempa{%
2663     \MT@map@clist@MT@features{{\MT@set@default@set{##1}{#2}}}%
2664   }%
2665   \MT@map@clist@MT@tempa{%
2666     \KV@sp@def\@tempa{##1}%
2667     \MT@ifempty\@tempa\relax{%
2668       \MT@is@feature{declaration of default set `#2'}{%
2669         \MT@exp@one@MT@set@default@set
2670         {\csname MT@rbba@\@tempa\endcsname}{#2}%
2671       }%
2672     }%
2673   }}%
2674   }%
2675 }

```

`\MT@default@pr@set`

```

\MT@default@ex@set 2676 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 2677   \KV@sp@def\@tempa{#2}%
2678   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@default@sp@set 2679     \MT@xdef@n{MT@default@#1@set}{\@nameuse{MT@abbr@#1} set `@\@tempa'}}%
\MT@default@kn@set 2680   \MT@xdef@n{MT@default@#1@set}{\@tempa}%
\MT@set@default@set 2681   }%
2682   \MT@error{%

```

```

2683     The \@nameuse{MT@abbr@#1} set '\@tempa' is not declared.\MessageBreak
2684     Cannot make it the default set. Using set\MessageBreak `all' instead{}}%
2685     \MT@xdef@n{MT@default@#1@set}{all}%
2686 }%
2687 }

```

### 14.3.2 Variants and aliases

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

```

\MT@variants
2688 \let\MT@variants\@empty
2689 \def\DeclareMicrotypeVariants{%
2690   \ifstar
2691     \MT@DeclareVariants
2692   {\let\MT@variants\@empty\MT@DeclareVariants}%
2693 }

```

**\MT@DeclareVariants**

```

2694 \def\MT@DeclareVariants#1{%
2695   \MT@map@clist@n{#1}%
2696   \KV@sp@def\@tempa{##1}%
2697   \@onelevel@sanitize\@tempa
2698   \xdef\MT@variants{\MT@variants{\@tempa}}%
2699 }%
2700 }

```

**\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.

```

2701 \renewcommand*\DeclareMicrotypeAlias[2]{%
2702   \KV@sp@def\@tempa{#1}%
2703   \KV@sp@def\@tempb{#2}%
2704   \@onelevel@sanitize\@tempb
2705   \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2706     \MT@warning{Alias font family '\@tempb' will override
2707       alias '\@nameuse{MT@\@tempa @alias}'\MessageBreak
2708       for font family '\@tempa'}}%
2709   \MT@xdef@n{MT@\@tempa @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.

```

2710   \MT@ifdefined@c@T{MT@family}%
2711   <debug>\MT@info{1}{Activating alias font '\@tempb' for '\MT@family'}%
2712   \MT@gl@et\MT@familyalias\@tempb
2713 }%
2714 }

```

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

```

2715 \def\LoadMicrotypeFile#1{%
2716   \KV@sp@def\@tempa{#1}%
2717   \@onelevel@sanitize\@tempa
2718   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2719   \ifMT@inlist@
2720     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2721   \else
2722     \MT@xadd\MT@file@list{\@tempa,}%
2723     \MT@begin@catcodes
2724     \InputIfFileExists{mt-\@tempa.cfg}{%
2725       \edef\MT@curr@file{mt-\@tempa.cfg}%
2726       \MT@vinfo{... Loading configuration file \MT@curr@file}%

```

```

2727 }{%
2728 \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2729             does not exist}%
2730 }%
2731 \MT@end@catcodes
2732 \fi
2733 }

```

### 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 2734 \MT@requires@pdftex5{
2735 \def\DisableLigatures{%
2736 \MT@begin@catcodes
2737 \MT@DisableLigatures
2738 }
2739 \newcommand*\MT@DisableLigatures[2][]{%
2740 \MT@ifempty{#1}\relax{\gdef\MT@n1@ligatures{#1}}%
2741 \xdef\MT@active@features{\MT@active@features,nl}%
2742 \global\MT@noligaturestrue
2743 \MT@declare@sets{nl}{no ligatures}{#2}%
2744 \gdef\MT@n1@setname{no ligatures}%
2745 \MT@end@catcodes
2746 }
2747 }{

```

If pdf<sub>T</sub><sub>E</sub>X is too old, we throw an error.

```

2748 \renewcommand*\DisableLigatures[2][]{%
2749 \MT@error{Disabling ligatures of a font is only possible\MessageBreak
2750           with pdftex version 1.30 or newer.\MessageBreak
2751           Ignoring \string\DisableLigatures}{Upgrade pdftex.}%
2752 }
2753 }

```

### 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.

```

2754 \def\DeclareMicrotypeBabelHook#1#2{%
2755 \MT@map@clist@n{#1}{%
2756 \KV@esp@def\@tempa{##1}%
2757 \MT@gdef@n{\MT@babel@{\@tempa}{#2}}%
2758 }%
2759 }

```

### 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).

```

2760 \def\SetProtrusion{%

```

```

2761 \MT@begin@catcodes
2762 \MT@SetProtrusion
2763 }

\MT@SetProtrusion    We want the catcodes to be correct even if this is called in the preamble.
\MT@pr@c@name 2764 \newcommand*\MT@SetProtrusion[3] [] {%
\MT@extra@context 2765 \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.
2766 \MT@set@named@keys{MT@pr@c}{#1}%
2767 debug\MT@edinfo{1}{creating protrusion list ` \MT@pr@c@name'%}
2768 \def\MT@permutelist{pr@c}%
2769 \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}, ...
2770 \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.
2771 \MT@gdefn{MT@pr@c@MT@pr@c@name}{#3}%
2772 \MT@end@catcodes
2773 }

\SetExpansion    \SetExpansion only differs in that it allows some extra options (stretch, shrink,
                    step, auto).
2774 \def\SetExpansion{%
2775 \MT@begin@catcodes
2776 \MT@SetExpansion
2777 }

\MT@SetExpansion
\MT@ex@c@name 2778 \newcommand*\MT@SetExpansion[3] [] {%
\MT@extra@context 2779 \let\MT@extra@context\@empty
2780 \MT@set@named@keys{MT@ex@c}{#1}%
\MT@permutelist 2781 \MT@ifdefinedn@T{MT@ex@c@MT@ex@c@name @factor}{%
2782 \ifnum\csname MT@ex@c@MT@ex@c@name @factor\endcsname > \@m
2783 \MT@warningn1{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
2784 too large in list\MessageBreak ` \MT@ex@c@name'. Setting it to the
2785 maximum of 1000}%
2786 \MT@glletnc{MT@ex@c@MT@ex@c@name @factor}\@m
2787 \fi
2788 }%
2789 debug\MT@edinfo{1}{creating expansion list ` \MT@ex@c@name'%}
2790 \def\MT@permutelist{ex@c}%
2791 \setkeys{MT@cfg}{#2}%
2792 \MT@permute
2793 \MT@gdefn{MT@ex@c@MT@ex@c@name}{#3}%
2794 \MT@end@catcodes
2795 }

\SetTracking
2796 \def\SetTracking{%
2797 \MT@begin@catcodes
2798 \MT@SetTracking
2799 }

\MT@SetTracking    Third argument may be empty.
2800 \newcommand*\MT@SetTracking[3] [] {%
2801 \let\MT@extra@context\@empty

```

```

2802 \MT@set@named@keys{MT@tr@}{#1}%
2803 (debug)\MT@info{1}{creating tracking list `\'MT@tr@name'}%
2804 \def\MT@permute{tr@}%
2805 \setkeys{MT@cfg}{#2}%
2806 \MT@permute
2807 \KV@sp@def\@tempa{#3}%
2808 \MT@ifempty\@tempa\relax{%
2809   \MT@ifint\@tempa
2810   {\MT@gdef\MT@tr@{\'MT@tr@name}{\'@tempa}}%
2811   {\MT@warning{Value `\'@tempa' is not a number in\MessageBreak
2812     tracking set `\'MT@curr@set@name'}}}%
2813 \MT@end@catcodes
2814 }

```

\SetExtraSpacing

```

2815 \def\SetExtraSpacing{%
2816   \MT@begin@catcodes
2817   \MT@SetExtraSpacing
2818 }

```

\MT@SetExtraSpacing

```

\MT@sp@c@name 2819 \newcommand*\MT@SetExtraSpacing[3][]{%
\MT@extra@context 2820 \let\MT@extra@context\empty
\MT@permute{list 2821 \MT@set@named@keys{MT@sp@c}{#1}%
2822 (debug)\MT@info{1}{creating spacing list `\'MT@sp@c@name'}%
2823 \def\MT@permute{sp@c}%
2824 \setkeys{MT@cfg}{#2}%
2825 \MT@permute
2826 \MT@gdef\MT@sp@c@{\'MT@sp@c@name}{#3}%
2827 \MT@end@catcodes
2828 }

```

\SetExtraKerning

```

2829 \def\SetExtraKerning{%
2830   \MT@begin@catcodes
2831   \MT@SetExtraKerning
2832 }

```

\MT@SetExtraKerning

```

\MT@kn@c@name 2833 \newcommand*\MT@SetExtraKerning[3][]{%
\MT@extra@context 2834 \let\MT@extra@context\empty
\MT@permute{list 2835 \MT@set@named@keys{MT@kn@c}{#1}%
2836 (debug)\MT@info{1}{creating kerning list `\'MT@kn@c@name'}%
2837 \def\MT@permute{kn@c}%
2838 \setkeys{MT@cfg}{#2}%
2839 \MT@permute
2840 \MT@gdef\MT@kn@c@{\'MT@kn@c@name}{#3}%
2841 \MT@end@catcodes
2842 }

```

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

\MT@options

```

2843 \def\MT@set@named@keys#1#2{%
2844   \def\x##1name=##2,##3\@nil{%
2845     \setkeys{#1}{name=##2}%
2846     \gdef\MT@options{##1##3}%
2847     \MT@rem@from@clist{name=}\MT@options
2848   }%
2849   \x#2,name=,\@nil
2850   \@expandtwoargs\setkeys{#1}\MT@options
2851 }

```

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

```
2852 \def\MT@define@code@key#1#2{%
2853   \define@key{MT@#2}{#1}[]{}%
2854   \@tempcnta=\@ne
2855   \MT@map@clist@n{##1}{%
2856     \KV@@sp@def\MT@val{###1}%
```

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

```
2857   \MT@get@highlevel{#1}%
2858   \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
2859   \advance\@tempcnta \@ne
2860 }%
2861 }%
2862 }
```

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

```
2863 \def\MT@define@code@key@size#1{%
2864   \define@key{MT@#1}{size}[]{}%
2865   \MT@map@clist@n{##1}{%
2866     \KV@@sp@def\MT@val{###1}%
2867     \expandafter\MT@get@range\MT@val--\@nil
2868     \ifx\MT@val\relax \else
2869       \MT@exp@cs\MT@xadd{MT@tempsize}%
2870       {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2871     \fi
2872   }%
2873 }%
2874 }
```

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

```
2875 \def\MT@define@code@key@font#1{%
2876   \define@key{MT@#1}{font}[]{}%
2877   \MT@map@clist@n{##1}{%
2878     \KV@@sp@def\MT@val{###1}%
2879     \MT@ifstreql\MT@val*{\def\MT@val{*/**/*/*}}\relax
2880     \expandafter\MT@get@font@and@size\MT@val///// \@nil
2881     \MT@xdef@n{MT@MT@permutelist @\@tempb\MT@extra@context}%
2882     {\csname MT@MT@permutelist @name\endcsname}%
2883     <debug>\MT@edinfo@n1{1}{initialising: use list for font \@tempb=\MT@val
2884     <debug> \ifx\MT@extra@context\@empty\else\MessageBreak
2885     <debug> (context: \MT@extra@context)\fi}%
2886     \MT@exp@cs\MT@xaddb
2887     {MT@MT@permutelist @\@tempb\MT@extra@context @size}%
2888     {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
2889   }%
2890 }%
2891 }
```

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

```
2892 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2893   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
2894 }

2895 \MT@define@code@key{encoding}{cfg}
2896 \MT@define@code@key{family}{cfg}
2897 \MT@define@code@key{series}{cfg}
2898 \MT@define@code@key{shape}{cfg}
2899 \MT@define@code@key@size{cfg}
2900 \MT@define@code@key@font{cfg}
```

`\MT@define@opt@key`

```

2901 \def\MT@define@opt@key#1#2{%
2902   \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
2903     \MT@xdefn{MT@#1@c@MT@curr@set@name @#2}{##1}}}%
2904 }

```

The options in the optional first argument.

```

2905 \MT@map@c@list@c\MT@features{%

```

Use file name and line number as the list name if the user didn't bother to invent one.

```

2906   \define@key{MT@#1@c}{name}[]{%
2907     \MT@ifempty{##1}{%
2908       \MT@edefn{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2909     }{%
2910       \MT@edefn{MT@#1@c@name}{##1}%
2911       \MT@ifdefinedn@T{MT@#1@c@csname MT@#1@c@name\endcsname}{%
2912         \MT@warning{Redefining \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}'}%
2913       }%
2914     }%
2915     \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
2916   }%
2917   \MT@define@opt@key{#1}{load}%
2918   \MT@define@opt@key{#1}{factor}%
2919   \MT@define@opt@key{#1}{preset}%
2920   \MT@define@opt@key{#1}{inputenc}%

```

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

```

2921   \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
2922 }

```

Automatically enable font copying if we find a protrusion or expansion context. After the preamble, check whether font copying is enabled. For older pdf<sub>T</sub><sub>E</sub>X versions, disallow. Also disable for lua<sub>T</sub><sub>E</sub>X.

```

2923 \MT@requires@pdftex7{
2924 (*lua)
2925   \MT@requires@lualatex{
2926     \define@key{MT@ex@c}{context}[]{%
2927       \MT@error{Expansion contexts currently don't work with luatex.\MessageBreak
2928         Ignoring `context' key\on@line}%
2929       {Use pdftex instead.}%
2930     }
2931   }{
2932 (/lua)
2933   \define@key{MT@ex@c}{context}[]{%
2934     \MT@ifempty{#1}\relax{%
2935       \MT@gl@et\MT@copy@font\MT@copy@font@
2936       \def\MT@extra@context{#1}%
2937     }%
2938   }
2939   \MT@addto@setup{%
2940     \define@key{MT@ex@c}{context}[]{%
2941       \ifx\MT@copy@font\MT@copy@font@
2942         \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
2943       \else
2944         \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
2945           Ignoring `context' key\on@line}%
2946         {Either move the settings inside the preamble,\MessageBreak
2947           or load the package with the `copyfonts' option.}%
2948       \fi
2949     }
2950   }

```

Protrusion contexts *may* 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.

```

2951 \define@key{MT@pr@c}{context}[]{%
2952   \MT@ifempty{#1}\relax{%
2953     \MT@glet\MT@copy@font\MT@copy@font@
2954     \def\MT@extra@context{#1}%
2955   }%
2956 }
2957 \MT@addto@setup{%
2958   \define@key{MT@pr@c}{context}[]{%
2959     \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
2960     \ifx\MT@copy@font\MT@copy@font@else
2961       \MT@warning@n{If protrusion contexts don't work as expected,
2962         \MessageBreak load the package with the `copyfonts' option}%
2963     \fi
2964   }
2965 }
2966 \lua }
2967 }{
2968   \define@key{MT@ex@c}{context}[]{%
2969     \MT@error{Expansion contexts only work with pdfTeX 1.40.4\MessageBreak
2970       or later. Ignoring `context' key\on@line}%
2971     {Upgrade pdfTeX.}%
2972   }
2973 }

\MT@warn@nodim
2974 \def\MT@warn@nodim#1{%
2975   \MT@warning{\`@tempa' is not a dimension.\MessageBreak
2976     Ignoring it and setting values relative to\MessageBreak #1}%
2977 }

```

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

```

2978 \define@key{MT@pr@c}{unit}[character]{%
2979   \MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
2980   \def\@tempa{#1}%
2981   \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.

```

2982   \MT@ifdimen\@tempa
2983   {\MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
2984   {\MT@warn@nodim{character widths}}%
2985 }%
2986 }

```

Tracking may only be relative to a dimension.

```

2987 \define@key{MT@tr@c}{unit}[1em]{%
2988   \MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
2989   \def\@tempa{#1}%
2990   \MT@ifdimen\@tempa
2991   {\MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
2992   {\MT@warn@nodim{1em}%
2993     \MT@gdef@nc{MT@tr@c@MT@curr@set@name @unit}{1em}}%
2994 }

```



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

```

2995 \MT@map@clist@n{sp,kn}{%
2996   \define@key{MT@#1@c}{unit}[space]{%
2997     \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
2998     \def\@tempa{##1}%
2999     \MT@ifstreq\@tempa{character}\relax{%
3000       \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
3001       \MT@ifstreq\@tempa{space}\relax{%
3002         \MT@ifdimen\@tempa
3003         {\MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
3004         {\MT@warn@nodim{width of space}}}%
3005       }%
3006     }%
3007   }%
3008 }

```

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

```

3009 \MT@map@clist@n{stretch,shrink,step}{%
3010   \define@key{MT@ex@c}{#1}[]{%
3011     \MT@ifempty{##1}\relax%
3012     \MT@ifint{##1}{%

```

A space terminates the number.

```

3013       \MT@gdef@n{MT@ex@c@MT@curr@set@name @#1}{##1 }%
3014     }{%
3015       \MT@warning{%
3016         Value `##1' for option `#1' is not a number.\MessageBreak
3017         Ignoring it}%
3018     }%
3019   }%
3020 }%
3021 }
3022 \define@key{MT@ex@c}{auto}[true]{%
3023   \def\@tempa{#1}%
3024   \csname if\@tempa\endcsname

```

Don't use `autoexpand` for pdf<sub>T</sub>E<sub>X</sub> version older than 1.20.

```

3025   \MT@requires@pdftex4{%
3026     \MT@gdef@n{MT@ex@c@MT@curr@set@name @auto}{autoexpand}%
3027   }{%
3028     \MT@warning{pdftex too old for automatic font expansion}%
3029   }
3030 \else
3031   \MT@requires@pdftex4{%
3032     \MT@glet@nc{MT@ex@c@MT@curr@set@name @auto}\@empty
3033   }\relax
3034 \fi
3035 }

```

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

```

3036 \MT@define@opt@key{tr}{spacing}
3037 \MT@define@opt@key{tr}{outerspacing}
3038 \MT@define@opt@key{tr}{outerkerning}

```

Which ligatures should be disabled?

```

3039 \define@key{MT@tr@c}{noligatures}[]%
3040   {\MT@xdef@n{MT@tr@c@MT@curr@set@name @noligatures}{#1}}
3041 \define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
3042 \define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
3043 \define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{noligatures={#1}}}

```

### 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.

```
3044 \renewcommand*\DeclareCharacterInheritance[1] [] {%
3045   \let\MT@extra@context\@empty
3046   \let\MT@extra@inputenc\@undefined
3047   \let\MT@inh@feat\@empty
3048   \setkeys{MT@inh@}{#1}%
3049   \MT@begin@catcodes
3050   \MT@set@inh@list
3051 }
```

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

```
3052 \def\MT@set@inh@list#1#2{%
3053   \MT@ifempty\MT@inh@feat{%
3054     \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
3055   }{%
3056     \MT@map@clist@c\MT@inh@feat{{%
3057       \KV@sp@def\@tempa{##1}%
3058       \MT@ifempty\@tempa\relax{%
3059         \MT@exp@one@n\MT@declare@char@inh
3060         {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
3061       }%
3062     }}%
3063   }%
3064   \MT@end@catcodes
3065 }
```

The keys for the optional argument.

```
3066 \MT@map@clist@c\MT@features@long{%
3067   \define@key{MT@inh@}{#1} [] {\edef\MT@inh@feat{\MT@inh@feat#1,}}
3068   \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
```

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

```
3069 \def\MT@declare@char@inh#1#2#3{%
3070   \MT@edef@n{MT@#1@inh@name}%
3071   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3072   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3073   \MT@ifdefined@c@T\MT@extra@inputenc{%
3074     \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3075   (debug)\MT@info{1}{creating inheritance list \@nameuse{MT@#1@inh@name}}%
3076   \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
3077   \def\MT@permutelist{#1@inh}%
3078   \setkeys{MT@inh@}{#2}%
3079   \MT@permute
3080 }
```

Parse the second argument. `\DeclareCharacterInheritance` may also be set up for various combinations.

```
3081 \define@key{MT@inh@}{encoding} [] {%
3082   \def\MT@val{#1}%
3083   \expandafter\MT@encoding@check\MT@val,\@nil
```

```

3084 \MT@get@highlevel{encoding}%
3085 \MT@edef@n{MT@tempencoding1}{\MT@val}%
3086 }

```

\MT@encoding@check But we only allow *one* encoding.

```

3087 \def\MT@encoding@check#1,#2\@nil{%
3088 \MT@ifempty{#2}\relax{%
3089 \edef\MT@val{#1}%
3090 \MT@warning{You may only specify one encoding for character\MessageBreak
3091 inheritance lists. Ignoring encoding(s) #2}%
3092 }%
3093 }

```

For the rest, we can reuse the key setup from the configuration lists (\Set...).

```

3094 \MT@define@code@key{family}{inh}
3095 \MT@define@code@key{series}{inh}
3096 \MT@define@code@key{shape}{inh}
3097 \MT@define@code@key{size}{inh}
3098 \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).

```

3099 \def\MT@inh@do#1,{%
3100 \ifx\relax#1\@empty \else
3101 \MT@inh@split #1==\relax
3102 \expandafter\MT@inh@do
3103 \fi
3104 }

```

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

```

3105 \def\MT@inh@split#1=#2=#3\relax{%
3106 \def\@tempa{#1}%
3107 \ifx\@tempa\@empty \else
3108 \MT@get@slot
3109 \ifnum\MT@char > \m@ne
3110 \let\MT@val\MT@char
3111 \MT@map@clist@n{#2}{%
3112 \def\@tempa{##1}%
3113 \ifx\@tempa\@empty \else
3114 \MT@get@slot
3115 \ifnum\MT@char > \m@ne
3116 \MT@exp@cs\MT@xadd{MT@inh@MT@listname @\MT@val @}{\MT@char}}%
3117 \fi
3118 \fi
3119 }%
3120 <debug>\MT@edinfo@n1{2}{children of #1 (\MT@val):
3121 <debug> \@nameuse{MT@inh@MT@listname @\MT@val @}}%
3122 \fi
3123 \fi
3124 }

```

### 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 \MT@permute@@ \MT@permute@@@ \MT@permute@@@

be the expansion of `\MT@<list type>@name`, i. e., the name of the currently defined list. Size ranges are held in a separate macro called `\MT@<list type>@<font axes>@sizes`, which in turn contains the respective *<list name>s* attached to the ranges.

```

3125 \def\MT@permute{%
3126   \let\MT@cnt@encoding\@ne
3127   \MT@permute@

  Undefine commands for the next round.

3128   \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
3129   \MT@glet\MT@tempsize\@undefined
3130 }
3131 \def\MT@permute@{%
3132   \let\MT@cnt@family\@ne
3133   \MT@permute@@
3134   \MT@increment\MT@cnt@encoding
3135   \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
3136   \MT@permute@
3137 }
3138 \def\MT@permute@@{%
3139   \let\MT@cnt@series\@ne
3140   \MT@permute@@@
3141   \MT@increment\MT@cnt@family
3142   \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
3143   \MT@permute@@
3144 }
3145 \def\MT@permute@@@{%
3146   \let\MT@cnt@shape\@ne
3147   \MT@permute@@@@
3148   \MT@increment\MT@cnt@series
3149   \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
3150   \MT@permute@@@@
3151 }
3152 \def\MT@permute@@@@{%
3153   \MT@permute@@@@@
3154   \MT@increment\MT@cnt@shape
3155   \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
3156   \MT@permute@@@@
3157 }
```

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

```

3158 \def\MT@permute@@@@@{%
3159   \MT@permute@define(encoding)%
3160   \ifMT@document
3161     \ifx\MT@tempencoding\@empty \else
3162       \MT@ifdefined@n@TF{T@MT@tempencoding}\relax
3163       {\expandafter\expandafter\expandafter\@gobble}%
3164     \fi
3165   \fi
3166   \MT@permute@@@@@@
3167 }
```

`\MT@permute@@@@@@`

```

3168 \def\MT@permute@@@@@@{%
3169   \MT@permute@define{family}%
3170   \MT@permute@define{series}%
3171   \MT@permute@define{shape}%
3172   \edef\@tempa{\MT@tempencoding
3173               /\MT@tempfamily
3174               /\MT@tempseries
3175               /\MT@tempshape
```

```
3176          /\MT@ifdefined@c@T\MT@tempsize *}%
```

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

```
3177 \MT@ifstreq\@tempa{////}\relax{%
3178   \ifx\MT@tempencoding\@empty
3179     \MT@warning{%
3180       You have to specify an encoding for\MessageBreak
3181       \@nameuse{MT@abbr@MT@permutelist} list
3182       `@nameuse{MT@MT@permutelist @name}'.\MessageBreak
3183       Ignoring it}%
3184   \else
3185     \MT@ifdefined@c@TF\MT@tempsize{%
```

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

```
3186     \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3187       \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3188     }%
3189     \MT@exp@cs\MT@xaddb
3190     {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3191     \MT@tempsize
3192 (debug) \MT@info@n1{1}{initialising: use list for font \@tempa,\MessageBreak
3193 (debug)       sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3194 (debug)       @sizes\endcsname}%
3195   }{%
```

Only one list can apply to a given combination.

```
3196     \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3197       \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3198       \@nameuse{MT@MT@permutelist @name}' will override list\MessageBreak
3199       \@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3200       for font \@tempa'}%
3201     }%
3202 (debug) \MT@info@n1{1}{initialising: use list for font \@tempa
3203 (debug)       \ifx\MT@extra@context\@empty\else\MessageBreak
3204 (debug)       (context: \MT@extra@context)\fi}%
3205   }%
3206     \MT@xdef@n{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3207       {\csname MT@MT@permutelist @name\endcsname}%
3208     \fi
3209   }%
3210 }
```

\MT@permute@define     Define the commands.

```
3211 \def\MT@permute@define#1{%
3212   \@tempcnta=\csname MT@cnt@#1\endcsname\relax
3213   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3214   {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3215   {\MT@let@nc{MT@temp#1}\@empty}%
3216 }
```

\MT@permute@reset     Reset the commands.

```
3217 \def\MT@permute@reset#1{%
3218   \@tempcnta=\@ne
3219   \MT@loop
3220     \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3221     \advance\@tempcnta\@ne
3222     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3223     \iftrue
3224     \iffalse
3225   \MT@repeat
3226 }
```

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

```
3227 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}
```

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

```
3228 \def\MT@check@rlist@#1#2#3{%
3229   \def\@tempb{#1}%
3230   \def\@tempc{#2}%
3231   \MT@if@false
3232   \MT@exp@cs\MT@map@tlist@c
3233   {MT\MT@permutelist @\@tempa\MT@extra@context @sizes}%
3234   \MT@check@range
3235 }
```

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

```
3236 \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, <#2> and <#3> those of the existing range.

```
3237 \def\MT@check@range@#1#2#3{%
3238   \MT@ifdim{#2}=\m@ne{%
3239     \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
3240     \MT@ifdim\@tempb={#1}\MT@if@true\relax
3241   }{%
```

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

```
3242     \MT@ifdim\@tempb>{#1}\relax{%
3243       \MT@ifdim\@tempc>{#1}{%
3244         \MT@if@true
3245         \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3246       }\relax
3247     }%
3248   }%
3249 }{%
3250   \MT@ifdim\@tempc=\m@ne{%
```

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

```
3251   \MT@ifdim\@tempb<{#2}{%
3252     \MT@ifdim\@tempb<{#1}\relax\MT@if@true
3253   }\relax
3254 }{%
```

- Both items are ranges.

```
3255   \MT@ifdim\@tempb<{#2}{%
3256     \MT@ifdim\@tempc>{#1}{%
3257       \MT@if@true
3258       \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3259     }\relax
3260   }\relax
3261 }%
3262 }%
3263 \ifMT@if@
3264   \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
3265     \@nameuse{MT@\MT@permutelist @name}' will override\MessageBreak
3266     list ~#3' for font \@tempa,\MessageBreak size \@tempb}%
```

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

```
3267 \expandafter\MT@tlist@break
3268 \fi
3269 }
```

## 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 3270 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 3271 \newif\ifMT@opt@auto
3272 \newif\ifMT@opt@DVI
```

`\MT@optwarn@admissible` Some warnings.

```
3273 \def\MT@optwarn@admissible#1#2{%
3274 \MT@warning@n1{`#1' is not an admissible value for option\MessageBreak
3275 \#2'. Assuming `false'}%
3276 }
```

`\MT@optwarn@nan`

```
3277 /package
3278 plain\MT@requires@latex1{
3279 \def\MT@optwarn@nan#1#2{%
3280 \MT@warning@n1{Value `#1' for option `#2' is not a\MessageBreak number.
3281 Using default value of \number\@nameuse{MT@#2@default}}%
3282 }
3283 plain}\relax
3284 *package
```

`\MT@opt@def@set`

```
3285 \def\MT@opt@def@set#1{%
3286 \MT@ifdefined@n@TF{MT@#1@tempb @set@@\MT@val}{%
3287 \MT@xdef@n{MT@#1@tempb @setname}{\MT@val}%
3288 }{%
3289 \MT@xdef@n{MT@#1@tempb @setname}{\@nameuse{MT@default@#1@tempb @set}}%
3290 \MT@warning@n1{The #1 set `MT@val' is undeclared.\MessageBreak
3291 Using set `MT@#1@tempb @setname' instead}%
3292 }%
3293 }
```

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

```
3294 \MT@map@clist@n{protrusion,expansion}{%
3295 \define@key{MT}{#1}[true]{%
3296 \csname MT@opt@#1true\endcsname
3297 \MT@map@clist@n{##1}{%
3298 \KV@sp@def\MT@val{###1}%
3299 \MT@ifempty\MT@val\relax%
3300 \csname MT@#1true\endcsname
3301 \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3302 \MT@ifstreq\MT@val{true}\relax
3303 }%
3304 \MT@ifstreq\MT@val{false}{%
3305 \csname MT@#1false\endcsname
3306 }{%
3307 \MT@ifstreq\MT@val{compatibility}{%
3308 \MT@let@nc{MT@#1@tempb @level}\@ne
3309 }{%
3310 \MT@ifstreq\MT@val{nocompatibility}{%

```

```

3311         \MT@let@nc{MT@ \@tempb @level}\tw@
3312     }{%

```

If everything failed, it should be a set name.

```

3313         \MT@opt@def@set{#1}%
3314     }%
3315 }%
3316 }%
3317 }%
3318 }%
3319 }%
3320 }%
3321 }

```

activate is a shortcut for protrusion and expansion.

```

3322 \define@key{MT}{activate}[true]{%
3323     \setkeys{MT}{protrusion={#1}}%
3324     \setkeys{MT}{expansion={#1}}%
3325 }

```

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

```

3326 \MT@map@clist@n{spacing,kerning,tracking}{%
3327     \define@key{MT}{#1}[true]{%
3328         \MT@map@clist@n{##1}{%
3329             \KV@@sp@def\MT@val{###1}%
3330             \MT@ifempty\MT@val\relax{%
3331                 \csname MT@#1true\endcsname
3332                 \MT@ifstreq\MT@val{true}\relax
3333             }%
3334             \MT@ifstreq\MT@val{false}{%
3335                 \csname MT@#1false\endcsname
3336             }%
3337             \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3338             \MT@opt@def@set{#1}%
3339         }%
3340     }%
3341 }%
3342 }%
3343 }%
3344 }

```

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

```

3345 \def\MT@def@bool@opt#1#2{%
3346     \define@key{MT}{#1}[true]{%
3347         \def\@tempa{##1}%
3348         \MT@ifstreq\@tempa{true}\relax{%
3349             \MT@ifstreq\@tempa{false}\relax{%
3350                 \MT@optwarn@admissible{##1}{#1}%
3351                 \def\@tempa{false}%
3352             }%
3353         }%
3354         #2%
3355     }%
3356 }

```

Boolean options that only set the switch.

```

3357 \MT@map@clist@n{draft,selected,babel}{%
3358     \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
3359 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotru}

```



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

```

3360 \MT@def@bool@opt{DVIoutput}{%
3361   \csname if\@tempa\endcsname
3362     \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
3363   \pdfoutput\z@
3364   \else
3365     \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
3366   \pdfoutput\@ne
3367   \fi
3368 }
```

Setting the `defersetup` option to false will restore the old behaviour, where the setup took place at the time when the package was loaded. This is undocumented, since I would like to learn about the cases where this is necessary.

The only problem with the new deferred setup I can think of is when a box is being constructed inside the preamble and this box contains a font that is not loaded before the box is being used.

```

3369 \MT@def@bool@opt{defersetup}{%
3370   \csname if\@tempa\endcsname \else
3371     \AtEndOfPackage{%
3372       \MT@setup@
3373       \let\MT@setup@\@empty
3374       \let\MT@addto@setup\@firstofone
3375     }%
3376   \fi
3377 }
```

`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.

```

3378 \MT@requires@pdftex7{
3379   (*lua)
3380   \MT@requires@luatex{
3381     \MT@def@bool@opt{copyfonts}{%
3382       \csname if\@tempa\endcsname
3383         \MT@error{The `copyfonts' option doesn't work with luatex}
3384         {Use pdftex instead.}%
3385       \fi
3386     }
3387   }{
3388     (/lua)
3389     \MT@def@bool@opt{copyfonts}{%
3390       \csname if\@tempa\endcsname
3391         \MT@glet\MT@copy@font\MT@copy@font@
3392       \else
3393         \MT@glet\MT@copy@font\relax
3394       \fi
3395     }
3396     (lua) }
3397   }{
3398     \MT@def@bool@opt{copyfonts}{%
3399       \csname if\@tempa\endcsname
3400         \MT@error{The pdftex version you are using is too old\MessageBreak
3401           to use the `copyfonts' option}{Upgrade pdftex.}%
3402       \fi
3403     }
3404   }
```

`final` is the opposite to `draft`.

```
3405 \MT@def@bool@opt{final}{%
3406   \csname if\@tempa\endcsname
3407   \MT@draftfalse
3408   \else
3409   \MT@drafttrue
3410   \fi
3411 }
```

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

```
3412 \define@key{MT}{verbose}[true]{%
3413   \let\MT@vinfo\MT@info@n1
3414   \def\@tempa{#1}%
3415   \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
3416   \MT@ifstreq\@tempa{errors}{%
3417     \let\MT@warning \MT@warn@err
3418     \let\MT@warning@n1 \MT@warn@err
3419   }{%
3420   \let\MT@vinfo\@gobble
```

Cast warnings to the winds.

```
3421   \MT@ifstreq\@tempa{silent}{%
3422     \let\MT@warning \MT@info
3423     \let\MT@warning@n1 \MT@info@n1
3424   }{%
3425   \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
3426   }%
3427 }%
3428 }%
3429 }
```

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

```
3430 <package>
3431 <plain>\MT@requires@latex1{
3432 \MT@map@clist@n{%
3433 <package> stretch,shrink,step,%
3434 letterspace}{%
3435 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
3436 \def\@tempa{##1 }%
```

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

```
3437 \MT@ifint\@tempa
3438 {\MT@edef@n{MT@#1}{\@tempa}}%
3439 {\MT@optwarn@nan{##1}{#1}}%
3440 }%
3441 }
3442 <plain>\relax
3443 <*package>
```

`factor` will define the protrusion factor only.

```
3444 \define@key{MT}{factor}[\MT@factor@default]{%
3445 \def\@tempa{#1 }%
3446 \MT@ifint\@tempa
3447 {\edef\MT@pr@factor{\@tempa}}
3448 {\MT@optwarn@nan{#1}{factor}}%
3449 }
```

Unit for protrusion codes.

```
3450 \define@key{MT}{unit}[character]{%
3451 \def\@tempa{#1}%
```

```

3452 \MT@ifstreq\@tempa{character}\relax{%
3453 \MT@ifdimen\@tempa
3454 {\let\MT@pr@unit\@tempa}%
3455 {\MT@warning@nl{\@tempa' is not a dimension.\MessageBreak
3456 Ignoring it and setting values relative to\MessageBreak
3457 character widths}}}%
3458 }%
3459 }

```

#### 14.4.2 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  $\text{\TeX}$  systems are switching to the pdf $\text{\TeX}$  engine even for DVI output, so that the user might not even be aware of the fact that she's running pdf $\text{\TeX}$ .)

```

3460 \MT@protrusiontrue
3461 \ifnum\pdfoutput<\@ne \else

```

Also, we only enable expansion by default if pdf $\text{\TeX}$  can expand the fonts automatically.

```

3462 \MT@requires@pdftex4{
3463 \MT@expansiontrue
3464 \MT@autottrue
3465 }\relax
3466 \fi

```

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.

```

3467 \define@key{MT}{config}[]{\relax}
3468 \def\MT@get@config#1config=#2,#3\@nil{%
3469 \MT@ifempty{#2}%
3470 {\def\MT@config@file{\MT@MT.cfg}}%
3471 {\def\MT@config@file{#2.cfg}}%
3472 }
3473 \expandafter\expandafter\expandafter\MT@get@config
3474 \csname opt@\@currname.\@current\endcsname,config=,\@nil

```

Load the file.

```

3475 \IfFileExists{\MT@config@file}{%
3476 \MT@info@nl{Loading configuration file \MT@config@file}%
3477 \MT@begin@catcodes
3478 \let\MT@begin@catcodes\relax
3479 \let\MT@end@catcodes\relax
3480 \let\MT@curr@file\MT@config@file
3481 \input{\MT@config@file}%
3482 \endgroup
3483 }{\MT@warning@nl{%
3484 Could not find configuration file \MT@config@file!\MessageBreak
3485 This will almost certainly cause undesired results.\MessageBreak
3486 Please fix your installation}%
3487 }

```

If no default font set has been declared in the main configuration file, we use the (empty, possibly non-existent) 'all' set.

```

3488 \MT@map@clist@c\MT@features{%
3489 \MT@ifdefined@n@TF{MT@default@#1@set}\relax
3490 {\MT@gdef@n{MT@default@#1@set}{all}}%

```

```

3491 }
\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).
3492 \def\MT@check@active@set#1{%
3493   \MT@ifdefined@n@TF{MT@#1@setname}{%
3494     \MT@info@n1{Using \@nameuse{MT@abbr@#1} set ` \@nameuse{MT@#1@setname}' }%
3495   }{%
3496     \MT@glet@nn{MT@#1@setname}{MT@default@#1@set}%
3497     \MT@info@n1{Using default \@nameuse{MT@abbr@#1} set ` \@nameuse{MT@#1@setname}' }%
3498   }%
3499 }

```

### 14.4.3 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.

```

3500 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
3501   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3502   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
3503 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook

```

### 14.4.4 Changing options later

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

```

3504 \def\microtypesetup{\setkeys{MT}}
3505 \MT@addto@setup{\def\microtypesetup{\setkeys{MTX}}}
3506 \def\MT@define@optionX#1#2{%

```

```

3507 \define@key{MTX}{#1}[true]{%
3508 \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3509 \MT@map@clist@{##1}{%
3510 \KV@sp@def\MT@val{###1}%
3511 \MT@ifempty\MT@val\relax{%
3512 \@tempcnta=\m@ne
3513 \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.

```

3514 \MT@checksetup{#1}{%
3515 \@tempcnta=\csname MT@\@tempb @level\endcsname
3516 \MT@info{Enabling #1
3517 (level \number\csname MT@\@tempb @level\endcsname)}%
3518 }%
3519 }{%
3520 \MT@ifstreq\MT@val{false}{%
3521 \@tempcnta=\z@
3522 \MT@info{Disabling #1}%
3523 }{%
3524 \MT@ifstreq\MT@val{compatibility}{%
3525 \MT@checksetup{#1}{%
3526 \@tempcnta=\@ne
3527 \MT@let@nc{MT@\@tempb @level}\@ne
3528 \MT@info{Setting #1 to level 1}%
3529 }%
3530 }{%
3531 \MT@ifstreq\MT@val{nocompatibility}{%
3532 \MT@checksetup{#1}{%
3533 \@tempcnta=\tw@
3534 \MT@let@nc{MT@\@tempb @level}\tw@
3535 \MT@info{Setting #1 to level 2}%
3536 }%
3537 }{\MT@error{Value `\'MT@val\' for key `\'#1\' not recognised}
3538 {Use any of `\'true\'', `\'false\'', `\'compatibility\'' or
3539 `\'nocompatibility\''.}%
3540 }%
3541 }%
3542 }%
3543 }%
3544 \ifnum\@tempcnta>\m@ne
3545 #2\@tempcnta\relax
3546 \fi
3547 }%
3548 }%
3549 }%
3550 }

```

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

```

3551 \def\MT@checksetup#1{%
3552 \csname ifMT@#1\endcsname
3553 \expandafter\@firstofone
3554 \else
3555 \MT@error{You cannot enable #1 if it was disabled\MessageBreak
3556 in the package options}{Load microtype with #1 enabled.}%
3557 \expandafter\@gobble
3558 \fi
3559 }

3560 \MT@define@optionX{protrusion}\pdfprotrudechars
3561 \MT@define@optionX{expansion}\pdfadjustspacing

```

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

```

3562 \MT@requires@pdftex6{
3563   \lua \MT@requires@luatex@firstofone{
3564     \def\MT@define@optionX@#1#2{%
3565       \define@key{MTX}{#1}[true]{%
3566         \MT@map@clist@n{##1}{%
3567           \KV@sp@def\MT@val{###1}%
3568           \MT@ifempty\MT@val\relax{%
3569             \@tempcnta=m@ne
3570             \MT@ifstreq\MT@val{true}{%
3571               \MT@checksetup{#1}{%
3572                 \@tempcnta=\@ne
3573                 \MT@info{Enabling #1}%
3574               }%
3575             }%
3576             \MT@ifstreq\MT@val{false}{%
3577               \@tempcnta=\z@
3578               \MT@info{Disabling #1}%
3579             }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
3580               {Use either `true' or `false'}}%
3581           }%
3582         }%
3583         \ifnum\@tempcnta>m@ne
3584           #2\relax
3585         \fi
3586       }%
3587     }%
3588   }%
3589 }

3590 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\relax
3591   \else\let\MT@tracking\MT@tracking@fi}
3592 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
3593 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3594   \pdfappendkern \@tempcnta}
3595 \gobble
3596 \lua }
3597 }\@firstofone

```

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

```

3598 {\define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
3599 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
3600 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
3601 }
3602 \define@key{MTX}{activate}[true]{%
3603   \setkeys{MTX}{protrusion={#1}}%
3604   \setkeys{MTX}{expansion={#1}}%
3605 }

```

`\MT@saved@setupfont` Disable everything – may be used as a work-around in case setting up fonts doesn't work in certain environments. (*Undocumented.*)

```

3606 \let\MT@saved@setupfont\MT@setupfont
3607 \define@key{MTX}{disable}[]{%
3608   \MT@info{Inactivate `~\MT@MT' package}%
3609   \let\MT@setupfont\relax
3610 }
3611 \define@key{MTX}{enable}[]{%
3612   \MT@info{Reactivate `~\MT@MT' package}%
3613   \let\MT@setupfont\MT@saved@setupfont
3614 }
3615 \package

```

### 14.4.5 Processing the options

`\MT@ProcessOptionsWithKV` Parse options.

```

3616 {plain}\MT@requires@latex1{
3617 \def\MT@ProcessOptionsWithKV#1{%
3618   \let\@tempc\relax
3619   \let\MT@temp\@empty
3620 {plain} \MT@requires@latex2{
3621   \MT@map@clist@c\@classoptionslist{%
3622     \def\CurrentOption{##1}%
3623     \MT@ifdefined@n@T{KV@#1@}\expandafter\MT@getkey\CurrentOption=\@nil}{%
3624       \edef\MT@temp{\MT@temp,\CurrentOption,}%
3625       \@expandtwoargs\@removeelement\CurrentOption
3626       \@unusedoptionlist\@unusedoptionlist
3627     }%
3628   }%
3629   \edef\MT@temp{\noexpand\setkeys{#1}%
3630     {\MT@temp\@optionlist{\@currname.\@currentext}}}%

```

`plain` can handle package options.

```

3631 {*plain}
3632   {\edef\MT@temp{\noexpand\setkeys{#1}%
3633     {\csname usepkg@options@usepkg@pkg\endcsname}}}
3634 {/plain}
3635   \MT@temp
3636   \MT@clear@options
3637 }

```

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

```

3638 \def\MT@getkey#1=#2\@nil{#1}
3639 \MT@ProcessOptionsWithKV{MT}
3640 {plain}\relax
3641 {*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).

```

3642 \MT@addto@setup{
3643 \ifMT@draft

```

We disable most of what we've just defined in the 3643 lines above if we are running in draft mode.

```

3644 \MT@warning@nl{'draft' option active.\MessageBreak
3645   Disabling all micro-typographic extensions.\MessageBreak
3646   This might lead to different line and page breaks}
3647 \let\MT@setupfont\relax
3648 \renewcommand*\LoadMicrotypeFile[1]{}
3649 \renewcommand*\microtypesetup[1]{}
3650 \renewcommand*\microtypecontext[1]{}
3651 \renewcommand*\lsstyle{}
3652 \else

```

For DVI output, the user must have explicitly passed the `expansion` option to the package.

```

3653 \ifnum\pdfoutput<\@ne
3654   \ifMT@opt@expansion \else
3655     \MT@expansionfalse
3656   \fi
3657 \fi

```

pdf<sub>T</sub><sub>E</sub>X 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!

```
3658 \MT@info@n1{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3659 \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
```

Fix the font sets.

```
3660 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
```

Protrusion.

```
3661 \ifMT@protrusion
3662 \edef\MT@active@features{\MT@active@features,pr}
3663 \pdfprotrudechars\MT@pr@level
3664 \MT@info@n1{Character protrusion enabled (level \number\MT@pr@level)}%
3665 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3666 factor: \number\MT@pr@factor\fi
3667 \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}
3668 \MT@check@active@set{pr}
3669 \else
3670 \let\MT@protrusion\relax
3671 \MT@info@n1{No character protrusion}
3672 \fi
```

Expansion.

```
3673 \ifMT@expansion
```

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

```
3674 \ifnum\MT@stretch=\m@ne
3675 \let\MT@stretch\MT@stretch@default
3676 \fi
```

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

```
3677 \ifnum\MT@shrink=\m@ne
3678 \let\MT@shrink\MT@stretch
3679 \fi
```

If `step` has not been specified, we will set it to  $\min(\text{stretch}, \text{shrink})/5$ , rounded off, minimum value 1.

```
3680 \ifnum\MT@step=\m@ne
3681 \ifnum\MT@stretch>\MT@shrink
3682 \ifnum\MT@shrink=\z@
3683 \@tempcnta=\MT@stretch
3684 \else
3685 \@tempcnta=\MT@shrink
3686 \fi
3687 \else
3688 \ifnum\MT@stretch=\z@
3689 \@tempcnta=\MT@shrink
3690 \else
3691 \@tempcnta=\MT@stretch
3692 \fi
3693 \fi
3694 \divide\@tempcnta 5\relax
```



```

3695 \else
3696 \tempcnta=\MT@step
3697 \ifnum\tempcnta=\z@
3698 \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
3699 Setting it to one}
3700 \fi
3701 \fi
3702 \ifnum\tempcnta=\z@ \tempcnta=\@ne \fi
3703 \edef\MT@step{\number\tempcnta\space}

```

`\MT@auto` Automatic expansion of the font? This new feature of pdf<sub>T</sub><sub>E</sub>X 1.20 makes the *l<sub>z</sub>* programme really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdf<sub>T</sub><sub>E</sub>X).

```

3704 \let\MT@auto\empty
3705 \ifMT@auto
3706 \MT@requires@pdftex4{%

```

We turn off automatic expansion if output mode is DVI.

```

3707 \ifnum\pdfoutput<\@ne
3708 \ifMT@opt@auto
3709 \MT@error{%
3710 Automatic font expansion only works for PDF output.\MessageBreak
3711 However, you are creating a DVI file}
3712 {If you have created expanded fonts instances, remove `auto' from%
3713 \MessageBreak the package options. Otherwise, you have to switch
3714 off expansion.\MessageBreak completely.}
3715 \fi
3716 \MT@autofalse
3717 \else
3718 \def\MT@auto{autoexpand}
3719 \fi

```

Also, if pdf<sub>T</sub><sub>E</sub>X is too old.

```

3720 }{%
3721 \MT@error{%
3722 The pdftex version you are using is too old for.\MessageBreak
3723 automatic font expansion}%
3724 {If you have created expanded fonts instances, remove `auto' from.\MessageBreak
3725 the package options. Otherwise, you have to switch off expansion.\MessageBreak
3726 completely, or upgrade pdftex to version 1.20 or newer.}
3727 \MT@autofalse
3728 \def\MT@auto{1000 }
3729 }
3730 \else

```

No automatic expansion.

```

3731 \MT@requires@pdftex4\relax{
3732 \def\MT@auto{1000 }
3733 }
3734 \fi

```

Choose the appropriate macro for selected expansion.

```

3735 \ifMT@selected
3736 \let\MT@set@ex@codes\MT@set@ex@codes@s
3737 \else
3738 \let\MT@set@ex@codes\MT@set@ex@codes@n
3739 \fi

```

Filter out stretch=0, shrink=0, since it would result in a pdf<sub>T</sub><sub>E</sub>X error.

```

3740 \ifnum\MT@stretch=\z@
3741 \ifnum\MT@shrink=\z@
3742 \MT@warning@nl{%

```

```

3743         Both the stretch and shrink limit are set to zero.\MessageBreak
3744         Disabling font expansion}
3745     \MT@expansionfalse
3746     \fi
3747     \fi
3748     \fi
3749     \ifMT@expansion
3750     \edef\MT@active@features{\MT@active@features,ex}%
3751     \pdfadjustspacing\MT@ex@level
3752     \MT@info@n1{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
3753         (level \number\MT@ex@level),\MessageBreak
3754         stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3755         step: \number\MT@step, \ifMT@selected\else non-\fi selected%
3756         \ifx\MT@copy@font\relax\else .\MessageBreak
3757         Using font copies for contexts\fi}

```

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

```

3758     \def\MT@check@step#1{%
3759         \@tempcnta=\csname MT@#1\endcsname
3760         \divide\@tempcnta \MT@step
3761         \multiply\@tempcnta \MT@step
3762         \ifnum\@tempcnta=\csname MT@#1\endcsname\else
3763             \MT@warning@n1{The #1 amount is not a multiple of step.\MessageBreak
3764                 The effective maximum #1 is \the\@tempcnta\space
3765                 (step \number\MT@step)}
3766         \fi
3767     }
3768     \MT@check@step{stretch}
3769     \MT@check@step{shrink}
3770     \MT@check@active@set{ex}

```

Inside \showhyphens, font expansion should be disabled.

```

3771     \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3772         \color@begingroup\everypar{}\parfillskip\z@skip
3773         \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3774         \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

```

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

```

3775     \gdef\showhyphens#1{\setbox0\vbox{%
3776         \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3777         \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3778         \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
3779     \else
3780         \let\MT@expansion\relax
3781         \MT@info@n1{No font expansion}
3782     \fi
3783 }
3784 \MT@requires@pdftex6{
3785 (*Lua)
3786 \def\MT@warn@lua#1{%
3787     \MT@error{The '~#1' feature doesn't currently work\MessageBreak with luatex}
3788     {Use pdftex instead.}%
3789     \MT@let@enc{MT@#1}\relax
3790 }
3791 
3792 
3793 \MT@addto@setup{%
3794 (*package)

```

Tracking, spacing and kerning.

```

3795     \ifMT@tracking
3796 (*Lua)     \MT@requires@luatex{\MT@warn@lua{tracking}}{

```

```

3797     \edef\MT@active@features{\MT@active@features,tr}
3798     \MT@info@nl{Tracking enabled}
3799     \MT@check@active@set{tr}

```

Enable protrusion for compensation at the line edges.

```

3800     \ifMT@protrusion\else\pdfprotrudechars\@ne\fi
3801   \lua
3802   \else
3803     \let\MT@tracking\relax
3804     \MT@info@nl{No tracking}
3805   \fi
3806   \ifMT@spacing
3807   \lua
3808     \MT@requires@luatex{\MT@warn@lua{spacing}}{
3809     \edef\MT@active@features{\MT@active@features,sp}
3810     \pdfadjustinterwordglue\@ne
3811     \MT@info@nl{Adjustment of interword spacing enabled}
3812     \MT@check@active@set{sp}
3813   }
3814   \else
3815     \let\MT@spacing\relax
3816     \MT@info@nl{No adjustment of interword spacing}
3817   \fi
3818   \ifMT@kerning
3819   \lua
3820     \MT@requires@luatex{\MT@warn@lua{kerning}}{
3821     \edef\MT@active@features{\MT@active@features,kn}
3822     \pdfprependkern\@ne
3823     \pdfappendkern\@ne
3824     \MT@info@nl{Adjustment of character kerning enabled}
3825     \MT@check@active@set{kn}
3826   }
3827   \else
3828     \let\MT@kerning\relax
3829     \MT@info@nl{No adjustment of character kerning}
3830   \fi
3831   \fi
3832   \fi
3833   \fi
3834   \fi
3835   \fi
3836   \fi
3837   \fi
3838   \fi
3839   \fi
3840   \fi
3841   \fi
3842   \fi
3843   \fi
3844   \fi
3845   \fi
3846   \fi
3847   \fi
3848   \fi
3849   \fi
3850   \fi

```

\MT@warn@tracking@DVI 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.

```

3830     \ifnum\pdfoutput<\@ne
3831     \def\MT@warn@tracking@DVI{%
3832     \MT@warning@nl{%
3833     You are using tracking/letterspacing in DVI mode.\MessageBreak
3834     This will probably not work, unless the post-\MessageBreak
3835     processing program (dvips, dvipdfm(x), ...) is\MessageBreak
3836     able to create the virtual fonts on the fly}%
3837     \MT@glet\MT@warn@tracking@DVI\relax
3838   }
3839   \else
3840     \def\MT@warn@tracking@DVI{%
3841     \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
3842     \MT@glet\MT@warn@tracking@DVI\relax
3843   }
3844   \fi
3845   \ifnum\MT@letterspace=\m@ne
3846     \let\MT@letterspace\MT@letterspace@default
3847   \else
3848     \MT@ls@too@large\MT@letterspace
3849   \fi
3850 }

```

If pdfTeX is too old, we disable tracking, spacing and kerning, and throw an error message.

```

3851 (*package)
3852 {}{
3853   \MT@addto@setup{%
3854     \ifMT@tracking
3855       \MT@error{Tracking only works with pdftex version 1.40\MessageBreak
3856         or newer. Switching it off}{Upgrade pdftex.}%
3857     \else
3858       \MT@info@nl{No tracking (pdftex too old)}
3859     \fi
3860     \ifMT@spacing
3861       \MT@error{Adjustment of interword spacing only works with\MessageBreak
3862         pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
3863     \else
3864       \MT@info@nl{No adjustment of interword spacing (pdftex too old)}
3865     \fi
3866     \ifMT@kerning
3867       \MT@error{Character kerning only works with\MessageBreak
3868         pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
3869     \else
3870       \MT@info@nl{No adjustment of character kerning (pdftex too old)}
3871     \fi
3872   }
3873 }
```

Warning if `\nonfrenchspacing` is active, since space factors will be ignored with `\pdfadjustinterwordglue>0`. Why 1500? Because some packages redefine `\frenchspacing`. See the c.t.t thread ‘`\frenchspacing` with AMS packages and babel’, started by Philipp Lehman on 16 August 2005: [ddtbaj\\$rob\\$1@online.de](mailto:ddtbaj$rob$1@online.de).

```

3874 \MT@requires@pdftex6{
3875   \AtBeginDocument{%
3876     \ifMT@spacing
3877       \ifMT@babel \else
3878         \ifnum\sfcodes>\. > 1500
3879           \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
3880             \MT@warning@nl{%
3881               \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3882               interword spacing will disable it. You might want\MessageBreak
3883               to add \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
3884               to your preamble}%
3885             }%
3886           \fi
3887         \fi
3888       \fi
3889     }
3890   }\relax
```

`\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```

3891 \MT@requires@pdftex5{
3892   \MT@addto@setup{%
3893     \ifMT@noligatures \else
3894       \let\MT@noligatures\relax
3895     \fi
3896   }
3897 }\relax
```

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

```

3898 \MT@addto@setup{%
3899   \ifx\MT@active@features\@empty \else
3900     \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3901   \fi
3902   \MT@documenttrue
3903 }

```

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

```

3904 \def\MT@set@babel@context#1{%
3905   \MT@ifdefined@n@TF{MT@babel@#1}{%
3906     \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
3907     \expandafter\MT@exp@one@n\expandafter\microtypecontext
3908     \csname MT@babel@#1\endcsname
3909   }{%
3910     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3911   }%
3912 }

```

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

```

3913 \@ifpackageloaded{babel}{
3914   \def\MT@shorthandoff#1#2{%
3915     \MT@info@n1{Switching off #1 babel's active characters (#2)}%
3916     \shorthandoff{#2}}
3917 }{
3918   \def\MT@shorthandoff#1#2{%
3919     \MT@error{You must load `babel' before `~\MT@MT'}
3920     {Otherwise, `~\MT@MT' cannot switch off #1 babel's\MessageBreak
3921     active characters.}}
3922 }

```

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

```

3923 \MT@addto@setup{%
3924   \ifMT@babel
3925     \@ifpackageloaded{babel}{%
3926       \MT@info@n1{Redefining babel's language switching commands}
3927       \let\MT@orig@select@language\select@language
3928       \def\select@language#1{%
3929         \MT@orig@select@language{#1}%
3930         \MT@set@babel@context{#1}%
3931       }
3932       \let\MT@orig@foreign@language\foreign@language
3933       \def\foreign@language#1{%
3934         \MT@orig@foreign@language{#1}%
3935         \MT@set@babel@context{#1}%
3936       }
3937       \ifMT@kerning

```

Disable French babel's active characters.

```

3938     \MT@if@false
3939     \MT@with@babel@and@T{french} \MT@if@true
3940     \MT@with@babel@and@T{frenchb} \MT@if@true
3941     \MT@with@babel@and@T{français}\MT@if@true
3942     \MT@with@babel@and@T{canadien}\MT@if@true
3943     \MT@with@babel@and@T{acadian} \MT@if@true
3944     \ifMT@if@MT@shorthandoff{French}{:;!}\fi

```

Disable Turkish babel's active characters.

```

3945     \MT@if@false
3946     \MT@with@babel@and@T{turkish} \MT@if@true
3947     \ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi
3948     \fi

```

In case babel was loaded before microtype:

```
3949 \MT@set@babel@context\languagename
3950 }{%
3951 \MT@warning@n{You did not load the babel package.\MessageBreak
3952 The `babel' option won't have any effect}
3953 }
3954 \fi
3955 }
```

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

```
3956 \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.

```
3957 \AtBeginDocument\selectfont
```

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

```
3958 \edef\MT@curr@file{\jobname.tex}
```

Restore catcodes.

```
3959 </package>
```

```
3960 \MT@restore@catcodes
```

That was that.

```
3961 </package|letterspace>
```

## 15 Configuration files

Let's now write the font configuration files.

```
3962 <*config>
```

```
3963
```

### 15.1 Font sets

We first declare some sets in the main configuration file.

```
3964 <*m-t>
3965 %%% -----
3966 %%% FONT SETS
3967
3968 \DeclareMicrotypeSet{all}
3969 { }
3970
3971 \DeclareMicrotypeSet{allmath}
3972 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
3973
3974 \DeclareMicrotypeSet{alltext}
3975 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
3976
3977 \DeclareMicrotypeSet{bascmath}
3978 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
3979   family   = {rm*,sf*},
3980   series    = {md*},
3981   size      = {normalsize,footnotesize,small,large}
3982 }
3983
3984 \DeclareMicrotypeSet{basictext}
```

```

3985 { encoding = {OT1,T1,LY1,OT4,QX,T5},
3986   family   = {rm*,sf*},
3987   series   = {md*},
3988   size     = {normalsize,footnotesize,small,large}
3989 }
3990
3991 \DeclareMicrotypeSet{smallcaps}
3992 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3993   shape    = {sc}
3994 }
3995
3996 \DeclareMicrotypeSet{footnotesize}
3997 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3998   size     = {-small}
3999 }
4000
4001 \DeclareMicrotypeSet{scriptsize}
4002 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
4003   size     = {-footnotesize}
4004 }
4005
4006 \DeclareMicrotypeSet{normal font}
4007 { font = */*/*/*/* }
4008

```

The default sets.

```

4009 %% -----
4010 %% DEFAULT SETS
4011
4012 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4013 \DeclareMicrotypeSetDefault[expansion]{basictext}
4014 \DeclareMicrotypeSetDefault[spacing]{basictext}
4015 \DeclareMicrotypeSetDefault[kerning]{alltext}
4016 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
4017

```

## 15.2 Font variants and aliases

```

4018 %% -----
4019 %% FONT VARIANTS AND ALIASES
4020

```

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

```

4021 \DeclareMicrotypeVariants{x,j,w,a,d,0,1}
4022

```

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 a variant, i. e., they shouldn't share a file.

Fonts that are 'the same': 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.

```

4023 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern

```

```

4024 \DeclareMicrotypeAlias{aer}{cmr} % ae
4025 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
4026 \DeclareMicrotypeAlias{cmor}{cmr} % eco
4027 \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`).

```

4028 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
4029 \DeclareMicrotypeAlias{qpl}{ppl} % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

```

The ‘FPL Neu’ fonts, a ‘re-implementation’ of Palatino.

```

4030 \DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
4031 \DeclareMicrotypeAlias{fp9j}{pplj} % "
4032 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
4033 \DeclareMicrotypeAlias{qtm}{ptm} % TeX Gyre Termes (formerly: qfonts/QuasiTimes)

```

More Times variants, to be checked: `pns`, `mns` (TimesNewRomanPS); `mnt` (TimesNewRomanMT, TimesNRSevenMT), `mtm` (TimesSmallTextMT); `pte` (TimesEuropa); `ptt` (TimesTen); TimesEighteen; TimesModernEF.

The `eulervm` package virtually extends the Euler fonts.

```

4034 \DeclareMicrotypeAlias{eur}{eur} % Euler VM
4035 \DeclareMicrotypeAlias{zeus}{eus} % "

```

MicroPress’s Charter version (`chmath`).

```

4036 \DeclareMicrotypeAlias{chr}{bch} % CH Math

```

The `mathdesign` package provides math fonts matching Bitstream Charter and URW Garamond.

```

4037 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
4038 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond

```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```

4039 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch

```

Euro symbol fonts, to save some files.

```

4040 \DeclareMicrotypeAlias{zpeus}{zpeu} % Adobe Euro sans -> serif
4041 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
4042 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
4043

```

## 15.3 Interaction with babel

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

```

4044 %% -----
4045 %% INTERACTION WITH THE `babel' PACKAGE
4046
4047 \DeclareMicrotypeBabelHook
4048 {english,UKenglish,british,USenglish,american}
4049 {kerning=, spacing=nonfrench}
4050
4051 \DeclareMicrotypeBabelHook
4052 {french,français,acadian,canadien}
4053 {kerning=french, spacing=}
4054
4055 \DeclareMicrotypeBabelHook
4056 {turkish}
4057 {kerning=turkish, spacing=}
4058

```



## 15.4 Note on admissible characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```
\ : \textbackslash
{ : \textbraceleft
} : \textbraceright
^ : \textasciicircum
% : \%
# : \#
```

Comma and equal sign must be guarded with braces ('{,}', '{=}') to keep keyval happy.

Character commands are allowed as far as they have been defined in the proper  $\text{\LaTeX}$  way, that is, when they have been assigned a slot in the font encoding with `\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.

## 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  $\text{\textcircled{C}}$  for  $\text{\textcircled{O}}$ .

```
4059 \m-t)
4060 \*m-t|zpeu|mys)
4061 %%% -----
4062 %%% CHARACTER INHERITANCE
4063
4064 \m-t|zpeu|mys)
4065 \*m-t)
```

### 15.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 ('fi' ligature), 013 ('fl'), 014 ('ffi'), 015 ('ffl'),  $\text{\textcircled{A}}$ ,  $\text{\textcircled{a}}$ ,  $\text{\textcircled{C}}$ ,  $\text{\textcircled{c}}$ .

```
4066 \DeclareCharacterInheritance
4067 { encoding = OT1 }
4068 { f = {011}, % ff
4069   i = {\i},
4070   j = {\j},
4071   0 = {\0},
4072   o = {\o}
4073 }
4074
```

### 15.5.2 T1

Candidates here: 028 (‘fi’), 029 (‘fl’), 030 (‘ffi’), 031 (‘ffl’), 156 (‘IJ’ ligature, since L<sup>A</sup>T<sub>E</sub>X 2005/12/01 accessible as \IJ), 188 (‘ij’, \ij), Æ, æ, Œ, œ.

```

4075 \DeclareCharacterInheritance
4076 { encoding = T1 }
4077 { A = {\`A,\'A,\^A,\-A,\"A,\r A,\k A,\u A},
4078   a = {\`a,\'a,\^a,\-a,\"a,\r a,\k a,\u a},
4079   C = {\`C,\'C,\^C,\-C,\"C,\r C},
4080   c = {\`c,\'c,\^c,\-c,\"c,\r c},
4081   D = {\`D,\'D,\^D,\-D,\"D,\r D},
4082   d = {\`d,\'d,\^d,\-d,\"d,\r d},
4083   E = {\`E,\'E,\^E,\-E,\"E,\r E},
4084   e = {\`e,\'e,\^e,\-e,\"e,\r e},
4085   f = {027}, % ff
4086   G = {\`G,\'G,\^G,\-G,\"G,\r G},
4087   g = {\`g,\'g,\^g,\-g,\"g,\r g},
4088   I = {\`I,\'I,\^I,\-I,\"I,\r I},
4089   i = {\`i,\'i,\^i,\-i,\"i,\r i},
4090   j = {\`j,\'j,\^j,\-j,\"j,\r j},
4091   L = {\`L,\'L,\^L,\-L,\"L,\r L},
4092   l = {\`l,\'l,\^l,\-l,\"l,\r l},
4093   N = {\`N,\'N,\^N,\-N,\"N,\r N},
4094   n = {\`n,\'n,\^n,\-n,\"n,\r n},
4095   O = {\`O,\'O,\^O,\-O,\"O,\r O},
4096   o = {\`o,\'o,\^o,\-o,\"o,\r o},
4097   R = {\`R,\'R,\^R,\-R,\"R,\r R},
4098   r = {\`r,\'r,\^r,\-r,\"r,\r r},
4099   S = {\`S,\'S,\^S,\-S,\"S,\r S},
4100   s = {\`s,\'s,\^s,\-s,\"s,\r s},
4101   T = {\`T,\'T,\^T,\-T,\"T,\r T},
4102   t = {\`t,\'t,\^t,\-t,\"t,\r t},
4103   U = {\`U,\'U,\^U,\-U,\"U,\r U},
4104   u = {\`u,\'u,\^u,\-u,\"u,\r u},
4105   Y = {\`Y,\'Y,\^Y,\-Y,\"Y,\r Y},
4106   y = {\`y,\'y,\^y,\-y,\"y,\r y},
4107   Z = {\`Z,\'Z,\^Z,\-Z,\"Z,\r Z},
4108   z = {\`z,\'z,\^z,\-z,\"z,\r z}

```

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

```

4109 % - = {127},
4110 }
4111

```

### 15.5.3 LY1

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

```

4112 \DeclareCharacterInheritance
4113 { encoding = LY1 }
4114 { A = {\`A,\'A,\^A,\-A,\"A,\r A},
4115   a = {\`a,\'a,\^a,\-a,\"a,\r a},
4116   C = {\`C,\'C,\^C,\-C,\"C,\r C},
4117   c = {\`c,\'c,\^c,\-c,\"c,\r c},
4118   D = {\`D,\'D,\^D,\-D,\"D,\r D},
4119   E = {\`E,\'E,\^E,\-E,\"E,\r E},
4120   e = {\`e,\'e,\^e,\-e,\"e,\r e},
4121   f = {011}, % ff
4122   I = {\`I,\'I,\^I,\-I,\"I,\r I},
4123   i = {\`i,\'i,\^i,\-i,\"i,\r i},

```

```

4124 L = {\L},
4125 l = {\l},
4126 N = {\-N},
4127 n = {\-n},
4128 O = {\^0,\'0,\^0,\-0,\"0,\0},
4129 o = {\^o,\'o,\^o,\-o,\"o,\o},
4130 S = {\v S},
4131 s = {\v s},
4132 U = {\^U,\'U,\^U,\"U},
4133 u = {\^u,\'u,\^u,\"u},
4134 Y = {\'Y,\"Y},
4135 y = {\'y,\"y},
4136 Z = {\v Z},
4137 z = {\v z}
4138 }
4139

```

#### 15.5.4 OT4

The Polish OT1 extension. More interesting characters here: 009 (‘fk’), 012 (‘fi’), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Ć, œ.

```

4140 \DeclareCharacterInheritance
4141 { encoding = OT4 }
4142 { A = {\k A},
4143   a = {\k a},
4144   C = {\'C},
4145   c = {\'c},
4146   E = {\k E},
4147   e = {\k e},
4148   f = {011}, % ff
4149   i = {\i},
4150   j = {\j},
4151   L = {\L},
4152   l = {\l},
4153   N = {\'N},
4154   n = {\'n},
4155   O = {\0,\'0},
4156   o = {\o,\'o},
4157   S = {\'S},
4158   s = {\'s},
4159   Z = {\'Z,\"Z},
4160   z = {\'z,\"z}
4161 }
4162

```

#### 15.5.5 QX

The Central European QX encoding.<sup>12</sup> Ligatures: 009 (‘fk’), 012 (‘fi’), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Ć, œ.

```

4163 \DeclareCharacterInheritance
4164 { encoding = QX }
4165 { A = {\^A,\'A,\^A,\-A,\"A,\k A,\AA},
4166   a = {\^a,\'a,\^a,\-a,\"a,\k a,\aa},
4167   C = {\'C,\"C},
4168   c = {\'c,\"c},
4169   D = {\DH},
4170   E = {\^E,\'E,\^E,\"E,\k E},

```

<sup>12</sup> Thanks to *Maciej Eder*.

```

4171 e = {\^e,\^e,\^e,\^e,\k e},
4172 f = {011}, % ff
4173 I = {\^I,\^I,\^I,\^I,\k I},
4174 i = {\^i,\^i,\^i,\^i,\k i,\^i},
4175 j = {\j},
4176 L = {\L},
4177 l = {\l},
4178 N = {\^N,\~N},
4179 n = {\^n,\~n},
4180 o = {\^0,\^0,\^0,\^0,\^0,\^0},
4181 o = {\^0,\^0,\^0,\^0,\^0,\^0},

```

The Rumanian `\textcommabelow` accents are actually replacements for the `\c` variants, which had previously (and erroneously<sup>13</sup>) been included in QX encoding. They are still kept for backwards compatibility.

```

4182 S = {\S,\c S,\textcommabelow S,\v S},
4183 s = {\s,\c s,\textcommabelow s,\v s},
4184 T = {\c T,\textcommabelow T},
4185 t = {\c t,\textcommabelow t},
4186 u = {\~u,\'u,\^u,\^u,\k u},
4187 u = {\~u,\'u,\^u,\^u,\k u},
4188 Y = {\'Y,\"Y},
4189 y = {\'y,\"y},
4190 Z = {\'Z,\^Z,\v Z},
4191 z = {\'z,\^z,\v z},
4192 . = \textellipsis
4193 }
4194

```

### 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.

```

4195 \DeclareCharacterInheritance
4196 { encoding = T5 }
4197 { A = {\^A,\^A,\-A,\h A,\d A,\^A,\u A,
4198         \\\Acircumflex,\'\Acircumflex,\-\Acircumflex,\h\Acircumflex,\d\Acircumflex,
4199         \\\Abreve,\'\Abreve,\-\Abreve,\h\Abreve,\d\Abreve},
4200   a = {\^a,\^a,\-a,\h a,\d a,\^a,\u a,
4201         \\\acircumflex,\'\acircumflex,\-\acircumflex,\h\acircumflex,\d\acircumflex,
4202         \\\abreve,\'\abreve,\-\abreve,\h\abreve,\d\abreve},
4203   D = {DJ},
4204   d = {dj},
4205   E = {\^E,\^E,\-E,\h E,\d E,\^E,
4206         \\\Ecircumflex,\'\Ecircumflex,\-\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
4207   e = {\^e,\^e,\-e,\h e,\d e,\^e,
4208         \\\ecircumflex,\'\ecircumflex,\-\ecircumflex,\h\ecircumflex,\d\ecircumflex},
4209   I = {\^I,\^I,\-I,\h I,\d I},
4210   i = {\^i,\^i,\-i,\h i,\d i,\^i},
4211   O = {\^O,\^O,\-O,\h O,\d O,\^O,\horn O,
4212         \\\Ocircumflex,\'\Ocircumflex,\-\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
4213         \\\Ohorn,\'\Ohorn,\-\Ohorn,\h\Ohorn,\d\Ohorn},
4214   o = {\^o,\^o,\-o,\h o,\d o,\^o,\horn o,
4215         \\\ocircumflex,\'\ocircumflex,\-\ocircumflex,\h\ocircumflex,\d\ocircumflex,
4216         \\\ohorn,\'\ohorn,\-\ohorn,\h\ohorn,\d\ohorn},
4217   U = {\^U,\^U,\-U,\h U,\d U,\horn U,
4218         \\\Uhorn,\'\Uhorn,\-\Uhorn,\h\Uhorn,\d\Uhorn},
4219   u = {\^u,\^u,\-u,\h u,\d u,\horn u,
4220         \\\uhorn,\'\uhorn,\-\uhorn,\h\uhorn,\d\uhorn},

```

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

```

4221     Y = {\`Y,\`Y,\~Y,\h Y,\d Y},
4222     y = {\`y,\`y,\~y,\h y,\d y}
4223 }
4224
4225 </m-t>

```

### 15.5.7 Euro symbols

Make Euro symbols settings simpler.

```

4226 < *zpeu >
4227 \DeclareCharacterInheritance
4228 { encoding = U,
4229   family   = {zpeu,zpeus,eurosans} }
4230 { E = 128 }
4231
4232 </zpeu >
4233 < *mvs >
4234 \DeclareCharacterInheritance
4235 { encoding = OT1,
4236   family   = mvs }
4237 { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
4238

```

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.

```

4239 \DeclareCharacterInheritance
4240 { encoding = U,
4241   family   = mvs }
4242 { 164 = {099,100,101} }
4243
4244 </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.

```

4245 < *m-t >
4246 %%% -----
4247 %%% TRACKING/LETTERSPACING
4248
4249 \SetTracking
4250 [ name      = default,
4251   no ligatures = {f} ]
4252 { encoding   = {OT1,T1,LY1,OT4,QX} }
4253 { }
4254

```

## 15.7 Font expansion

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

```

4255 %%% -----
4256 %%% EXPANSION
4257
4258 \SetExpansion
4259 [ name      = default      ]

```

```

4260 { encoding = {OT1,OT4,QX,T1,LY1} }
4261 {
4262     A = 500,      a = 700,
4263     \AE = 500,    \ae = 700,
4264     B = 700,      b = 700,
4265     C = 700,      c = 700,
4266     D = 500,      d = 700,
4267     E = 700,      e = 700,
4268     F = 700,
4269     G = 500,      g = 700,
4270     H = 700,      h = 700,
4271     K = 700,      k = 700,
4272     M = 700,      m = 700,
4273     N = 700,      n = 700,
4274     O = 500,      o = 700,
4275     \OE = 500,    \oe = 700,
4276     P = 700,      p = 700,
4277     Q = 500,      q = 700,
4278     R = 700,
4279     S = 700,      s = 700,
4280     U = 700,      u = 700,
4281     W = 700,      w = 700,
4282     Z = 700,      z = 700,
4283     2 = 700,
4284     3 = 700,
4285     6 = 700,
4286     8 = 700,
4287     9 = 700
4288 }
4289

```

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

```

4290 \SetExpansion
4291 [ name      = T5 ]
4292 { encoding = T5 }
4293 {
4294     A = 500,      a = 700,
4295     B = 700,      b = 700,
4296     C = 700,      c = 700,
4297     D = 500,      d = 700,
4298     E = 700,      e = 700,
4299     F = 700,
4300     G = 500,      g = 700,
4301     H = 700,      h = 700,
4302     K = 700,      k = 700,
4303     M = 700,      m = 700,
4304     N = 700,      n = 700,
4305     O = 500,      o = 700,
4306     P = 700,      p = 700,
4307     Q = 500,      q = 700,
4308     R = 700,
4309     S = 700,      s = 700,
4310     U = 700,      u = 700,
4311     W = 700,      w = 700,
4312     Z = 700,      z = 700,
4313     2 = 700,
4314     3 = 700,
4315     6 = 700,
4316     8 = 700,
4317     9 = 700
4318 }
4319
4320 </m-t>

```

## 15.8 Character protrusion

```
4321 %%% -----
4322 %%% PROTRUSION
4323
```

For future historians, Hàn Thế Thành's original settings (from `protcode.tex`, converted to microtype notation).

```
\SetProtrusion
[ name      = thanh ]
{ encoding = OT1 }
{
  A = {50,50},
  F = { ,50},
  J = {50, },
  K = { ,50},
  L = { ,50},
  T = {50,50},
  V = {50,50},
  W = {50,50},
  X = {50,50},
  Y = {50,50},
  k = { ,50},
  r = { ,50},
  t = { ,50},
  v = {50,50},
  w = {50,50},
  x = {50,50},
  y = {50,50},
  . = { ,700},    {,}= { ,700},
  : = { ,500},    ; = { ,500},
  ! = { ,200},    ? = { ,200},
  ( = {50, },    ) = { ,50},
  - = { ,700},
  \textendash     = { ,300},    \textemdash     = { ,200},
  \textquoteleft  = {700, },    \textquoteright = { ,700},
  \textquotedblleft = {500, }, \textquotedblright = { ,500}
}
```

### 15.8.1 Default

The default settings always use the most moderate value.

```
4324 <*cfg-t>
4325 \SetProtrusion
4326 <m-t> [ name      = default ]
```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code `bch`)

```
4327 <bch> [ name      = bch-default ]
```

- Bitstream Letter Gothic (`blg`)

```
4328 <blg> [ name      = blg-default ]
```

- Computer Modern Roman (`cmr`)

```
4329 <cmr> [ name      = cmr-default ]
```

- Adobe Garamond (`pad`, `padx`, `padj`)

```
4330 <pad> [ name      = pad-default ]
```

- Minion<sup>14</sup> (pmnx, pmnj)

```
4331 <pmn> [ name = pmnj-default ]
```

- Palatino (ppl, pplx, pplj)

```
4332 <ppl> [ name = ppl-default ]
```

- Times (ptm, ptmx, ptmj)

```
4333 <ptm> [ name = ptm-default ]
```

- URW Garamond (ugm)

```
4334 <ugm> [ name = ugm-default ]
4335 <m-t> { encoding = OT1 }
4336 <cmr> { }
4337 <bch|blg|pad|pmn|ugm> { encoding = OT1,
4338 <ppl|ptm> { encoding = {OT1,OT4},
4339 <bch> family = bch }
4340 <blg> family = blg }
4341 <pad> family = {pad,padx,padj }
4342 <pmn> family = pmnj }
4343 <ppl> family = {ppl,pplx,pplj }
4344 <ptm> family = {ptm,ptmx,ptmj }
4345 <ugm> family = ugm }
4346 {
4347 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> A = {50,50},
4348 <ugm> A = {50,100},
4349 <m-t|pad|ptm> \AE = {50, },
4350 <ugm> \AE = {150,50},
4351 <ugm> B = { ,50},
4352 <bch|pad|pmn|ugm> C = {50, },
4353 <bch|pad|pmn> D = { ,50},
4354 <ugm> D = { ,70},
4355 <ugm> E = { ,50},
4356 <m-t|bch|cmr|pad|pmn|ptm> F = { ,50},
4357 <ugm> F = { ,70},
4358 <bch|pad|pmn> G = {50, },
4359 <ugm> G = {50,50},
4360 <blg> I = {150,150},
4361 <m-t|cmr|pad|pmn|ppl|ptm|ugm> J = {50, },
4362 <bch|blg> J = {100, },
4363 <!blg> K = { ,50},
4364 <blg> K = {50, },
4365 <m-t|bch|cmr|pad|pmn|ppl> L = { ,50},
4366 <blg> L = { ,150},
4367 <ptm> L = { ,80},
4368 <ugm> L = { ,120},
4369 <bch|pad|pmn|ugm> O = {50,50},
4370 <pad|pmn> \OE = {50, },
4371 <ugm> \OE = {50,50},
4372 <blg> P = { ,100},
4373 <ugm> P = { ,50},
4374 <bch|pad|pmn> Q = {50,70},
4375 <ugm> Q = {50,50},
4376 <bch> R = { ,50},
4377 <ugm> R = { ,70},
4378 <m-t|bch|cmr|pad|pmn|ppl|ptm> T = {50,50},
4379 <blg> T = {100,100},
4380 <ugm> T = {70,70},
4381 <m-t|bch|cmr|pad|pmn|ppl|ptm> V = {50,50},
```

---

14 Contributed by Harald Harders ([h.harders@tu-bs.de](mailto:h.harders@tu-bs.de)).



```

4382 <blg|ugm>      V = {70,70},
4383 <m-t|bch|cmr|pad|pmn|ppl|ptm>      W = {50,50},
4384 <ugm>      W = {70,70},
4385 <m-t|bch|cmr|pad|pmn|ppl|ptm>      X = {50,50},
4386 <ugm>      X = {50,70},
4387 <m-t|bch|cmr|pad|pmn|ppl>      Y = {50,50},
4388 <blg|ptm|ugm>      Y = {80,80},
4389 <ugm>      Z = {50,50},
4390 <blg>      f = {150,100},
4391 <blg>      i = {150,150},
4392 <blg>      j = {100,100},
4393 <m-t|bch|cmr|pad|pmn|ppl|ptm>      k = { ,50},
4394 <ugm>      k = { ,70},
4395 <blg>      l = {150,150},
4396 <pmn>      l = { ,50},
4397 <pad|ppl>      p = {50,50},
4398 <ugm>      p = { ,50},
4399 <pad|ppl>      q = {50, },
4400 <!blg>      r = { ,50},
4401 <blg>      r = {100, 80},
4402 <cmr|pad|pmn>      t = { ,70},
4403 <bch>      t = { ,50},
4404 <blg>      t = {150, 80},
4405 <ugm>      t = { ,100},
4406 <m-t|bch|cmr|pad|pmn|ppl|ptm>      v = {50,50},
4407 <blg>      v = {100,100},
4408 <ugm>      v = {50,70},
4409 <m-t|bch|cmr|pad|pmn|ppl|ptm>      w = {50,50},
4410 <ugm>      w = {50,70},
4411 <!blg>      x = {50,50},
4412 <blg>      x = {100,100},
4413 <m-t|bch|pad|pmn>      y = { ,50},
4414 <blg>      y = { 50,100},
4415 <cmr|ppl|ptm>      y = {50,70},
4416 <ugm>      y = { ,70},

4417 <cmr>      0 = { ,50},
4418 <m-t>      1 = {50,50},
4419 <bch|blg|pad|ptm|ugm>      1 = {150,150},
4420 <cmr>      1 = {100,200},
4421 <pmn>      1 = { ,50},
4422 <ppl>      1 = {100,100},
4423 <bch|cmr|pad|ugm>      2 = {50,50},
4424 <blg>      2 = { ,100},
4425 <bch|pmn>      3 = {50, },
4426 <cmr|pad|ugm>      3 = {50,50},
4427 <blg>      3 = {100, },
4428 <m-t|pad>      4 = {50,50},
4429 <bch>      4 = {100,50},
4430 <blg>      4 = {100, },
4431 <cmr|ugm>      4 = {70,70},
4432 <pmn>      4 = {50, },
4433 <ptm>      4 = {70, },
4434 <cmr>      5 = { ,50},
4435 <pad>      5 = {50,50},
4436 <bch>      6 = {50, },
4437 <cmr>      6 = { ,50},
4438 <pad>      6 = {50,50},
4439 <m-t>      7 = {50,50},
4440 <bch|pad|pmn|ugm>      7 = {50,80},
4441 <blg>      7 = {100,100},
4442 <cmr|ptm>      7 = {50,100},
4443 <ppl>      7 = { ,50},

```

```

4444 <cmr>      8 = { ,50},
4445 <bch|pad>   9 = {50,50},
4446 <cmr>      9 = { ,50},
4447 <m-t|cmr|pad|pmn|ppl|ptm|ugm> . = { ,700},
4448 <bch>       . = { ,600},
4449 <blg>       . = {400,500},
4450 <!blg>     {,}= { ,500},
4451 <blg>       {,}= {300,400},
4452 <m-t|cmr|pad|pmn|ppl|ptm|ugm> : = { ,500},
4453 <bch>       : = { ,400},
4454 <blg>       : = {300,400},
4455 <m-t|bch|pad|pmn|ptm> ; = { ,300},
4456 <blg>       ; = {200,300},
4457 <cmr|ppl>   ; = { ,500},
4458 <ugm>       ; = { ,400},
4459 <!blg>     ! = { ,100},
4460 <blg>       ! = {200,200},
4461 <m-t|pad|pmn|ptm> ? = { ,100},
4462 <bch|cmr|ppl|ugm> ? = { ,200},
4463 <blg>       ? = {150,150},
4464 <pmn>       " = {300,300},
4465 <m-t|bch|cmr|pad|pmn|ppl> @ = {50,50},
4466 <ptm>       @ = {100,100},
4467 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> ~ = {200,250},
4468 <ugm>       ~ = {300,350},
4469 <pad|ppl|ptm> & = {50,100},
4470 <ugm>       & = { ,100},
4471 <m-t|cmr|pad|pmn> \% = {50,50},
4472 <bch>       \% = { ,50},
4473 <ppl|ptm>   \% = {100,100},
4474 <ugm>       \% = {50,100},
4475 <blg>       \# = {100,100},
4476 <m-t|ppl|ptm|ugm> * = {200,200},
4477 <bch|pmn>   * = {200,300},
4478 <blg>       * = {150,200},
4479 <cmr|pad>   * = {300,300},
4480 <m-t|cmr|ppl|ptm> + = {250,250},
4481 <bch>       + = {150,250},
4482 <pad>       + = {300,300},
4483 <blg|pmn>   + = {150,200},
4484 <ugm>       + = {250,300},
4485 <blg|ugm>   {=}= {200,200},
4486 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
4487 <bch|ugm>   ( = {200, }, ) = { ,200},
4488 <cmr|blg>   ( = {300, }, ) = { ,300},
4489 <ppl>       ( = {100, }, ) = { ,300},
4490 <bch|pmn>   [ = {100, }, ] = { ,100},
4491 <blg>       [ = {300,100}, ] = { ,300},

4492 <m-t|pad|pmn|ptm> / = {100,200},
4493 <bch>       / = { ,200},
4494 <blg>       / = {300,300},
4495 <cmr|ppl>   / = {200,300},
4496 <ugm>       / = {100,300},
4497 <m-t|ptm>   - = {500,500},
4498 <bch|cmr|ppl> - = {400,500},
4499 <blg>       - = {300,400},
4500 <pad>       - = {300,500},
4501 <pmn>       - = {200,400},
4502 <ugm>       - = {500,600},
4503 <blg>       < = {200,100}, > = {100,200},
4504 <blg>       _ = {150,250},
4505 <blg>       | = {250,250},

```

```

4506 <m-t|pmn> \textendash = {200,200}, \textemdash = {150,150},
4507 <bch> \textendash = {200,300}, \textemdash = {150,250},
4508 <cmr> \textendash = {400,300}, \textemdash = {300,200},
4509 <pad|ppl|ptm> \textendash = {300,300}, \textemdash = {200,200},
4510 <ugm> \textendash = {250,300}, \textemdash = {250,250},

```

Why settings for left *and* right quotes? Because in some languages they might be used like that (see the csquotes package for examples).

```

4511 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
4512 <blg> \textquoteleft = {400,600}, \textquoteright = {400,600},
4513 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
4514 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
4515 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
4516 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
4517 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300}
4518 <blg> \textquotedblright = {300,400}
4519 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600}
4520 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4521 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400}
4522 }
4523

```

Greek uppercase letters are in OT1 encoding only.

```

4524 <*cmr>
4525 \SetProtrusion
4526 [ name = cmr-OT1,
4527   load = cmr-default ]
4528 { encoding = {OT1,OT4},
4529   family = cmr }
4530 {
4531   \AE = { 50, },
4532   "00 = { ,150}, % \Gamma
4533   "01 = {100,100}, % \Delta
4534   "02 = { 50, 50}, % \Theta
4535   "03 = {100,100}, % \Lambda
4536   "06 = { 50, 50}, % \Sigma
4537   "07 = {100,100}, % \Upsilon
4538   "08 = { 50, 50}, % \Phi
4539   "09 = { 50, 50} % \Psi

```

Remaining slots can be found in the source file.

```

4540 }
4541
4542 </cmr>

```

T1 and LY1 encodings contain some more characters. The default list will be loaded first.

```

4543 \SetProtrusion
4544 <m-t> [ name = T1-default,
4545 <bch> [ name = bch-T1,
4546 <blg> [ name = blg-T1,
4547 <cmr> [ name = cmr-T1,
4548 <pad> [ name = pad-T1,
4549 <pmn> [ name = pmnj-T1,
4550 <ppl> [ name = ppl-T1,
4551 <ptm> [ name = ptm-T1,
4552 <ugm> [ name = ugm-T1,
4553 <m-t> load = default ]
4554 <bch> load = bch-default ]
4555 <blg> load = blg-default ]
4556 <cmr> load = cmr-default ]
4557 <pad> load = pad-default ]

```

```

4558 <pmn>    load      = pmnj-default ]
4559 <ppl>     load      = ppl-default ]
4560 <ptm>     load      = ptm-default ]
4561 <ugm>     load      = ugm-default ]
4562 <m-t>     { encoding = {T1,LY1} }
4563 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4564 <blg|ptm|ugm> { encoding = {T1},
4565 <bch>      family   = bch }
4566 <blg>      family   = blg }
4567 <cmr>      family   = cmr }
4568 <pad>      family   = {pad,padx,padj} }
4569 <pmn>      family   = pmnj }
4570 <ppl>      family   = {ppl,pplx,pplj} }
4571 <ptm>      family   = {ptm,ptmx,ptmj} }
4572 <ugm>      family   = ugm }
4573 {
4574 <cmr>      \AE = {50, },
4575 <bch>      \OE = {50, },
4576 <pmn>      \TH = { ,50},
4577 <blg>      \v L = { ,250},
4578 <blg>      \v d = { ,250},
4579 <blg>      \v l = { ,250},
4580 <blg>      \v t = { ,250},
4581 <blg>      127 = {300,400},
4582 <blg>      156 = {100, }, % IJ
4583 <blg>      188 = { 80, 80}, % ij
4584 <m-t|bch|pad|pmn|ppl|ptm> - = {100,100},
4585 <cmr>      - = {200,200},
4586 <ugm>      - = {100,200},
4587 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
4588 <bch>      \textbackslash = {150,200},
4589 <blg>      \textbackslash = {250,300},
4590 <cmr|ppl> \textbackslash = {200,300},
4591 <ugm>      \textbackslash = {100,300},
4592 <ugm>      \textbar      = {200,200},
4593 <blg>      \textendash    = {300,300}, \textemdash      = {150,150},
4594 <blg>      \textquotedbl = {300,400}, \textquotedblleft = {300,400},
4595 <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.

```

4596 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4597 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
4598 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4599 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4600 <blg> \guilsinglleft = {300,500}, \guilsinglright = {300,500},
4601 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4602 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4603 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
4604 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
4605 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
4606 <blg|pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
4607 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4608 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
4609 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
4610 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4611 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
4612 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
4613 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
4614 <pmn> \textless = {100, }, \textgreater = { ,100},
4615 <pmn> \textvisiblespace = {100,100} % not in LY1
4616 }

```

4617

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

```
4618 (*cmr)
4619 \SetProtrusion
4620 [ name      = lmr-T1,
4621   load      = cmr-T1 ]
4622 { encoding = {T1,LY1},
4623   family   = lmr      }
4624 {
4625   \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4626 }
4627
4628 (/cmr)
```

Settings for the QX encoding (generic and Times). It also includes some glyphs otherwise in TS1.

```
4629 (*m-t|ptm)
4630 \SetProtrusion
4631 <m-t> [ name      = QX-default,
4632 <ptm> [ name      = ptm-QX,
4633 <m-t>   load      = default ]
4634 <ptm>   load      = ptm-default ]
4635 <m-t> { encoding = QX }
4636 <ptm> { encoding = QX,
4637 <ptm>   family   = {ptm,ptmx,ptmj} }
4638 {
4639 <ptm>   * = {200,200},
4640   {=} = {100,100},
4641   \textunderscore = {100,100},
4642   \textbackslash   = {100,200},
4643   \quotedblbase    = {400,400},
4644 <m-t>   \guillemotleft = {200,200}, \guillemotright = {200,200},
4645 <ptm>   \guillemotleft = {300,300}, \guillemotright = {200,400},
4646   \textexclamdown = {100, }, \textquestiondown = {100, },
4647 <m-t>   \textbraceleft = {400,200}, \textbraceright = {200,400},
4648 <ptm>   \textbraceleft = {200,200}, \textbraceright = {200,300},
4649   \textless        = {200,100}, \textgreater       = {100,200},
4650   \textminus       = {200,200}, \textdegree      = {300,300},
4651 <m-t>   \copyright    = {100,100}, \textregistered = {100,100}
4652 <ptm>   \copyright    = {100,150}, \textregistered = {100,150},
4653 <ptm>   \textxgeq     = { ,100}, \textxleq      = {100, },
4654 <ptm>   \textalpha    = { , 50}, \textDelta     = { 70, 70},
4655 <ptm>   \textpi       = { 50, 80}, \textSigma    = { , 70},
4656 <ptm>   \textmu       = { , 80}, \texteuro      = { 50, 50},
4657 <ptm>   \textellipsis = {150,200}, \textasciitilde = { 80, 80},
4658 <ptm>   \textapprox   = { 50, 50}, \textinfty    = {100,100},
4659 <ptm>   \textdagger   = {150,150}, \textdaggerdbl = {100,100},
4660 <ptm>   \textdiv      = { 50,150}, \textsection  = { 80, 80},
4661 <ptm>   \texttimes    = {100,150}, \textpm       = { 50, 80},
4662 <ptm>   \textbullet   = {150,150}, \textperiodcentered = {300,300},
4663 <ptm>   \textquotesingle = {500,500}, \textquotedbl = {300,300},
4664 <ptm>   \textperthousand = { ,50}
4665 }
4666
4667 (/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.

```
4668 (*cmr|bch)
4669 \SetProtrusion
```

```

4670 <cmr> [ name      = cmr-T5,
4671 <cmr>   load       = cmr-default ]
4672 <bch> [ name      = bch-T5,
4673 <bch>   load       = bch-default ]
4674 { encoding = T5,
4675 <cmr>   family    = cmr }
4676 <bch>   family    = bch }
4677 {
4678 <bch>   _ = {100,100},
4679 <bch>   \textbackslash = {150,200},
4680 <cmr>   \textbackslash = {200,300},
4681 <cmr>   \textquotedblleft = {200,600},
4682 <cmr>   \textquotedbl = {300,300},
4683 <bch>   \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4684 <cmr>   \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4685 <bch>   \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4686 <cmr>   \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4687 <bch>   \guillemotleft = {200,200}, \guillemotright = {150,300},
4688 <cmr>   \guillemotleft = {300,200}, \guillemotright = {100,400},
4689 <bch>   \textbraceleft = {200,  }, \textbraceright = {  ,300},
4690 <cmr>   \textbraceleft = {400,200}, \textbraceright = {200,400},
4691   \textless = {200,100}, \textgreater = {100,200}
4692 }
4693
4694 </cmr|bch>
4695 <*pmn>
4696 \SetProtrusion
4697 [ name      = pmnx-OT1,
4698   load      = pmnj-default ]
4699 { encoding = OT1,
4700   family    = pmnx }
4701 {
4702   1 = {230,180}
4703 }
4704
4705 \SetProtrusion
4706 [ name      = pmnx-T1,
4707   load      = pmnj-T1 ]
4708 { encoding = {T1,LY1},
4709   family    = pmnx }
4710 {
4711   1 = {230,180}
4712 }
4713
4714 </pmn>

```

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

```

4715 <*ptm>
4716 \SetProtrusion
4717 [ name      = ptm-LY1,
4718   load      = ptm-T1 ]
4719 { encoding = LY1,
4720   family    = {ptm,ptmx,ptmj} }
4721 {
4722   _ = {100,100},
4723   \texttrademark = {100,100},
4724   \textregistered = {100,100},
4725   \textcopyright = {100,100},
4726   \textdegree = {300,300},
4727   \textminus = {200,200},
4728   \textellipsis = {150,200},

```

```

4729 % \texteuro = { , }, % ?
4730 \textcent = {100,100},
4731 \textquotesingle = {500,500},
4732 \textflorin = { 50, 70},
4733 \textdagger = {150,150},
4734 \textdaggerdbl = {100,100},
4735 \textperthousand = { , 50},
4736 \textbullet = {150,150},
4737 \textonesuperior = {100,100},
4738 \texttwosuperior = { 50, 50},
4739 \textthreesuperior = { 50, 50},
4740 \textperiodcentered = {300,300},
4741 \textplusminus = { 50, 80},
4742 \textmultiply = {100,100},
4743 \textdivide = { 50,150}

```

Remaining slots in the source file.

```

4744 }
4745
4746 </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.

Therefore, we leave the letters away, and only set up the punctuation characters.

```

4747 \SetProtrusion
4748 <m-t> [ name = OT1-it ]
4749 <bch> [ name = bch-it ]
4750 <blg> [ name = blg-it,
4751 <blg> load = blg-default ]
4752 <cmr> [ name = cmr-it ]
4753 <pad> [ name = pad-it ]
4754 <pmn> [ name = pmn-it ]
4755 <ppl> [ name = ppl-it ]
4756 <ptm> [ name = ptm-it ]
4757 <ugm> [ name = ugm-it ]
4758 <m-t|bch|blg|pad|pmn|ugm> { encoding = OT1,
4759 <ppl|ptm> { encoding = {OT1,OT4},
4760 <bch> family = bch,
4761 <blg> family = blg,
4762 <pad> family = {pad,padx,padj},
4763 <pmn> family = pmnj,
4764 <ppl> family = {ppl,pplx,pplj},
4765 <ptm> family = {ptm,ptmx,ptmj},
4766 <ugm> family = ugm,
4767 <m-t|bch|pad|pmn|ppl|ptm> shape = {it,sl} }
4768 <blg|ugm> shape = it }
4769 <cmr> { }
4770 {
4771 <cmr|ptm> A = {100,50},
4772 <pad|pmn> A = {50, },
4773 <ugm> A = { ,150},
4774 <ppl> A = {50,50},
4775 <ptm> \AE = {100, },
4776 <pad|ppl> \AE = {50, },
4777 <pmn> \AE = { , -50},
4778 <cmr|pad|ppl|ptm> B = {50, },
4779 <pmn> B = {20, -50},
4780 <bch|ppl|ptm|ugm> C = {50, },
4781 <cmr|pad> C = {100, },

```

```

4782 <pmn>      C = {50,-50},
4783 <cmr|pad|ppl|ptm> D = {50,50},
4784 <pmn>      D = {20, },
4785 <cmr|pad|ppl|ptm> E = {50, },
4786 <pmn>      E = {20,-50},
4787 <cmr|pad|ptm> F = {100, },
4788 <pmn>      F = {10, },
4789 <ppl>      F = {50, },
4790 <bch|ppl|ptm|ugm> G = {50, },
4791 <cmr|pad>   G = {100, },
4792 <pmn>      G = {50,-50},
4793 <cmr|pad|ppl|ptm> H = {50, },
4794 <cmr|pad|ptm> I = {50, },
4795 <pmn>      I = {20,-50},
4796 <cmr|ptm>   J = {100, },
4797 <pad>      J = {50, },
4798 <pmn>      J = {20, },
4799 <cmr|pad|ppl|ptm> K = {50, },
4800 <pmn>      K = {20, },
4801 <cmr|pad|ppl|ptm> L = {50, },
4802 <pmn>      L = {20,50},
4803 <ugm>      L = { ,100},
4804 <cmr|ptm>   M = {50, },
4805 <pmn>      M = { , -30},
4806 <cmr|ptm>   N = {50, },
4807 <pmn>      N = { , -30},
4808 <bch|pmn|ppl|ptm> O = {50, },
4809 <cmr|pad>   O = {100, },
4810 <ugm>      O = {70,50},
4811 <pmn|ppl|ptm> \OE = {50, },
4812 <pad>      \OE = {100, },
4813 <cmr|pad|ppl|ptm> P = {50, },
4814 <pmn>      P = {20,-50},
4815 <bch|pmn|ppl|ptm> Q = {50, },
4816 <cmr|pad>   Q = {100, },
4817 <ugm>      Q = {70,50},
4818 <cmr|pad|ppl|ptm> R = {50, },
4819 <pmn>      R = {20, },
4820 <bch|cmr|pad|ppl|ptm> S = {50, },
4821 <pmn>      S = {20,-30},
4822 <bch|cmr|pad|ppl|ptm> $ = {50, },
4823 <pmn>      $ = {20,-30},
4824 <bch|pmn|ugm> T = {70, },
4825 <cmr|pad|ppl|ptm> T = {100, },
4826 <cmr|pad|ppl|ptm> U = {50, },
4827 <pmn>      U = {50,-50},
4828 <cmr|pad|pmn|ugm> V = {100, },
4829 <ppl|ptm>   V = {100,50},
4830 <cmr|pad|pmn|ugm> W = {100, },
4831 <ppl>      W = {50, },
4832 <ptm>      W = {100,50},
4833 <cmr|ppl|ptm> X = {50, },
4834 <cmr|ptm>   Y = {100, },
4835 <pmn>      Y = {50, },
4836 <ppl>      Y = {100,50},
4837 <pmn>      Z = { , -50},
4838 <pmn>      d = { , -50},
4839 <pad|pmn>   f = { , -100},
4840 <pmn>      i = { , -30},
4841 <pmn>      j = { , -30},
4842 <pmn>      l = { , -100},
4843 <bch>      o = {50,50},
4844 <bch>      p = { , 50},

```



```

4845 <pmn>      p = {-50, },
4846 <bch>      q = {50, },
4847 <pmn>      r = { ,50},
4848 <bch>      t = { ,50},
4849 <pmn|ugm>   v = {50, },
4850 <bch>      w = { ,50},
4851 <pmn|ugm>   w = {50, },
4852 <bch>      y = { ,50},
4853 <cmr>      0 = {100, },
4854 <bch|ptm>   1 = {150,100},
4855 <cmr>      1 = {200,50},
4856 <pad>      1 = {150, },
4857 <pmn>      1 = {50, },
4858 <ppl>      1 = {100, },
4859 <ugm>      1 = {150,150},
4860 <cmr>      2 = {100,-100},
4861 <pad|ppl|ptm> 2 = {50, },
4862 <pmn>      2 = {-50, },
4863 <bch>      3 = {50, },
4864 <cmr>      3 = {100,-100},
4865 <pmn>      3 = {-100, },
4866 <ptm>      3 = {100,50},
4867 <bch>      4 = {100, },
4868 <cmr|pad>   4 = {150, },
4869 <ppl|ptm>   4 = {50, },
4870 <cmr>      5 = {100, },
4871 <ptm>      5 = {50, },
4872 <bch>      6 = {50, },
4873 <cmr>      6 = {100, },
4874 <bch|pad|ptm> 7 = {100, },
4875 <cmr>      7 = {200,-150},
4876 <pmn>      7 = {20, },
4877 <ppl>      7 = {50, },
4878 <cmr>      8 = {50,-50},
4879 <cmr>      9 = {100,-100},
4880 <m-t|cmr|pad|pmn|ppl> . = { ,500},
4881 <blg>      . = {400,600},
4882 <bch|ptm|ugm> . = { ,700},
4883 <blg>      {,}= {300,500},
4884 <m-t|cmr|pad|pmn|ppl> {,}= { ,500},
4885 <bch|ugm>   {,}= { ,600},
4886 <ptm>      {,}= { ,700},
4887 <m-t|cmr|pad|ppl>   := { ,300},
4888 <bch|ugm>   := { ,400},
4889 <pmn>      := { ,200},
4890 <ptm>      := { ,500},
4891 <m-t|cmr|pad|ppl>   ; = { ,300},
4892 <bch|ugm>   ; = { ,400},
4893 <pmn>      ; = { ,200},
4894 <ptm>      ; = { ,500},
4895 <ptm>      ! = { ,100},
4896 <bch>      ? = { ,200},
4897 <ptm>      ? = { ,100},
4898 <ppl>      ? = { ,300},
4899 <pmn>      " = {400,200},
4900 <m-t|pad|pmn|ppl|ptm> & = {50,50},
4901 <bch>      & = { ,80},
4902 <cmr>      & = {100,50},
4903 <ugm>      & = {50,100},
4904 <m-t|cmr|pad|pmn>   \% = {100, },
4905 <bch>      \% = {50,50},
4906 <ppl|ptm>   \% = {100,100},
4907 <ugm>      \% = {100,50},

```

```

4908 <m-t|pmn|ppl>      * = {200,200},
4909 <bch>      * = {300,200},
4910 <cmr>      * = {400,100},
4911 <pad>      * = {500,100},
4912 <ptm|ugm>      * = {400,200},
4913 <m-t|cmr|pmn|ppl>      + = {150,200},
4914 <bch|ugm>      + = {250,250},
4915 <pad|ptm>      + = {250,200},
4916 <m-t|pad|pmn|ppl>      @ = {50,50},
4917 <bch>      @ = {80,50},
4918 <cmr>      @ = {200,50},
4919 <ptm>      @ = {150,150},
4920 <m-t|bch|ugm>      ~ = {150,150},
4921 <cmr|pad|pmn|ppl|ptm>      ~ = {200,150},
4922 <ugm>      {=} = {200,200},
4923 <!blg>      ( = {200, }, ) = { ,200},
4924 <m-t|cmr|pad|ppl|ptm|ugm>      / = {100,200},
4925 <bch>      / = { ,150},
4926 <pmn>      / = {100,150},
4927 <m-t>      - = {300,300},
4928 <bch|pad>      - = {300,400},
4929 <pmn>      - = {200,300},
4930 <cmr>      - = {500,300},
4931 <ppl>      - = {300,500},
4932 <ptm>      - = {500,500},
4933 <ugm>      - = {400,700},
4934 <blg>      - = {0,300},
4935 <m-t|pmn>      \textendash = {200,200}, \textendash = {150,150},
4936 <bch>      \textendash = {200,300}, \textendash = {150,200},
4937 <cmr>      \textendash = {500,300}, \textendash = {400,200},
4938 <pad|ppl|ptm|ugm>      \textendash = {300,300}, \textendash = {200,200},
4939 <m-t|bch|pmn|ugm>      \textquoteleft = {400,200}, \textquoteright = {400,200},
4940 <blg>      \textquoteleft = {400,400}, \textquoteright = {400,400},
4941 <cmr|pad>      \textquotedblleft = {800,200}, \textquotedblright = {800,200},
4942 <ppl>      \textquotedblleft = {700,400}, \textquotedblright = {700,400},
4943 <ptm>      \textquotedblleft = {800,500}, \textquotedblright = {800,500},
4944 <m-t|bch|pmn>      \textquotedblleft = {400,200}, \textquotedblright = {400,200}
4945 <blg>      \textquotedblright = {300,300}
4946 <cmr>      \textquotedblleft = {700,100}, \textquotedblright = {500,300}
4947 <pad>      \textquotedblleft = {700,200}, \textquotedblright = {700,200}
4948 <ppl>      \textquotedblleft = {500,300}, \textquotedblright = {500,300}
4949 <ptm>      \textquotedblleft = {700,400}, \textquotedblright = {700,400}
4950 <ugm>      \textquotedblleft = {600,200}, \textquotedblright = {600,200}
4951 }
4952
4953 <*cmr>
4954 \SetProtrusion
4955 [ name = cmr-it-OT1,
4956   load = cmr-it ]
4957 { encoding = {OT1,OT4},
4958   family = cmr,
4959   shape = it }
4960 {
4961   \AE = {100, },
4962   \OE = {100, },
4963   "00 = {200,150}, % \Gamma
4964   "01 = {150,100}, % \Delta
4965   "02 = {150, 50}, % \Theta
4966   "03 = {150, 50}, % \Lambda
4967   "04 = {100,100}, % \Xi
4968   "05 = {100,100}, % \Pi
4969   "06 = {100, 50}, % \Sigma
4970   "07 = {200,150}, % \Upsilon

```

```

4971 "08 = {150, 50}, % \Phi
4972 "09 = {150,100}, % \Psi
4973 "0A = { 50, 50} % \Omega
4974 }
4975
4976 </cmr>
4977 \SetProtrusion
4978 <m-t> [ name = Tl-it-default,
4979 <bch> [ name = bch-it-Tl,
4980 <blg> [ name = blg-it-Tl,
4981 <cmr> [ name = cmr-it-Tl,
4982 <pad> [ name = pad-it-Tl,
4983 <pmn> [ name = pmn-it-Tl,
4984 <ppl> [ name = ppl-it-Tl,
4985 <ptm> [ name = ptm-it-Tl,
4986 <ugm> [ name = ugm-it-Tl,
4987 <m-t> load = OTl-it ]
4988 <bch> load = bch-it ]
4989 <blg> load = blg-Tl ]
4990 <cmr> load = cmr-it ]
4991 <pmn> load = pmnj-it ]
4992 <pad> load = pad-it ]
4993 <ppl> load = ppl-it ]
4994 <ptm> load = ptm-it ]
4995 <ugm> load = ugm-it ]
4996 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {Tl,LYl},
4997 <blg|ptm|ugm> { encoding = Tl,
4998 <bch> family = bch,
4999 <blg> family = blg,
5000 <cmr> family = cmr,
5001 <pmn> family = pmnj,
5002 <pad> family = {pad,padx,padj},
5003 <ppl> family = {ppl,pplx,pplj},
5004 <ptm> family = {ptm,ptmx,ptmj},
5005 <ugm> family = ugm,
5006 <m-t|bch|pad|pmn|ppl|ptm> shape = {it,sl} }
5007 <blg|cmr|ugm> shape = it }
5008 {
5009 <m-t|bch|pmn> _ = { ,100},
5010 <blg> _ = {0,300},
5011 <cmr|ugm> _ = {100,200},
5012 <pad|ppl|ptm> _ = {100,100},
5013 <blg> . = {400,600},
5014 <blg> {,}= {300,500},
5015 <cmr> \AE = {100, },
5016 <bch> \OE = { 50, },
5017 <cmr> \OE = {100, },
5018 <pmn> 031 = { , -100}, % ffl
5019 <cmr|ptm> 156 = {100, }, % IJ
5020 <pad> 156 = {50, }, % IJ
5021 <pmn> 156 = {20, }, % IJ
5022 <pmn> 188 = { , -30}, % ij
5023 <pmn> \v t = { ,100},
5024 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
5025 <cmr|ugm> \textbackslash = {300,300},
5026 <bch> \textbackslash = {150,150},
5027 <pmn> \textbackslash = {100,150},
5028 <ugm> \textbar = {200,200},
5029 <cmr> \textquotedblleft = {500,300},
5030 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
5031 <blg> \textquotedbl = {300,300}, \textquotedblleft = {300,300},
5032 <blg> \textquotedblright = {300,300}, \quotedblbase = {200,600},
5033 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},

```

```

5034 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5035 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
5036 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
5037 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
5038 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5039 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5040 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5041 <pad> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
5042 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
5043 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
5044 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
5045 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
5046 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
5047 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
5048 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
5049 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
5050 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
5051 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
5052 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
5053 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
5054 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
5055 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
5056 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
5057 <pmn> \textvisiblespace = {100,100}
5058 }
5059
5060 <*m-t|ptm>
5061 \SetProtrusion
5062 <m-t> [ name = QX-it-default,
5063 <ptm> [ name = ptm-it-QX,
5064 <m-t> load = OT1-it ]
5065 <ptm> load = ptm-it ]
5066 { encoding = {QX},
5067 <ptm> family = {ptm,ptmx,ptmj},
5068 shape = {it,sl} }
5069 {
5070 <ptm> 009 = { , 50}, % fk
5071 {=} = {100,100},
5072 <m-t> \textunderscore = {100,100},
5073 <ptm> \textunderscore = {100,150},
5074 \textbackslash = {100,200},
5075 \quotedblbase = {300,400},
5076 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
5077 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
5078 \textexclamdown = {200, }, \textquestiondown = {200, },
5079 \textbraceleft = {200,100}, \textbraceright = {200,200},
5080 \textless = {100,100}, \textgreater = {100,100},
5081 \textminus = {200,200}, \textdegree = {300,150},
5082 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5083 <ptm> \textregistered = {100,150}, \copyright = {100,150},
5084 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
5085 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
5086 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
5087 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
5088 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
5089 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
5090 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5091 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
5092 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
5093 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
5094 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5095 <ptm> \textperthousand = { ,50}
5096 }

```

```

5097
5098 </m-t|ptm>
5099 <*cmr|bch>
5100 \SetProtrusion
5101 <cmr> [ name = cmr-it-T5,
5102 <cmr>   load = cmr-it ]
5103 <bch> [ name = bch-it-T5,
5104 <bch>   load = bch-it ]
5105 { encoding = T5,
5106 <bch>   family = bch,
5107 <cmr>   family = cmr,
5108   shape = it }
5109 {
5110 <bch>   _ = { ,100},
5111 <cmr>   _ = {100,200},
5112 <bch>   \textbackslash = {150,150},
5113 <cmr>   \textbackslash = {300,300},
5114 <bch>   \quotesinglbase = {200,500}, \quotedblbase = {150,500},
5115 <cmr>   \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5116 <bch>   \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5117 <cmr>   \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5118 <bch>   \guillemotleft = {200,300}, \guillemotright = {150,400},
5119 <cmr>   \guillemotleft = {400,100}, \guillemotright = {200,300},
5120 <bch>   \textbraceleft = {200, }, \textbraceright = { ,200},
5121 <cmr>   \textbraceleft = {400,100}, \textbraceright = {200,200},
5122 <bch>   \textless = {100, }, \textgreater = { ,100},
5123 <cmr>   \textless = {300,100}, \textgreater = {200,100}
5124 }
5125
5126 </cmr|bch>

```

Slanted is very similar to italic.

```

5127 <*cmr>
5128 \SetProtrusion
5129 [ name = cmr-sl,
5130   load = cmr-it-OT1 ]
5131 { encoding = {OT1,OT4},
5132   family = cmr,
5133   shape = sl }
5134 {
5135   L = { ,50},
5136   f = { , -50},
5137   - = {300, },
5138   \textendash = {400, }, \textemdash = {300, }
5139 }
5140
5141 \SetProtrusion
5142 [ name = cmr-sl-T1,
5143   load = cmr-it-T1 ]
5144 { encoding = {T1,LY1},
5145   family = cmr,
5146   shape = sl }
5147 {
5148   L = { ,50},
5149   f = { , -50},
5150   - = {300, },
5151   \textendash = {400, }, \textemdash = {300, }
5152 }
5153
5154 \SetProtrusion
5155 [ name = cmr-sl-T5,
5156   load = cmr-it-T5 ]
5157 { encoding = T5,

```

```

5158     family    = cmr,
5159     shape      = sl }
5160 {
5161     L = { ,50},
5162     f = { ,-50},
5163     - = {300, },
5164     \textendash = {400, }, \textemdash = {300, }
5165 }
5166
5167 \SetProtrusion
5168 [ name = lmr-it-T1,
5169   load = cmr-it-T1 ]
5170 { encoding = {T1,LY1},
5171   family   = lmr,
5172   shape     = {it,sl} }
5173 {
5174   \textquotedblleft = { ,200}, \textquotedblright = { ,200},
5175   \quotesinglbase    = { ,400}, \quotedblbase     = { ,500}
5176 }
5177

```

Oldstyle numerals are slightly different.

```

5178 \SetProtrusion
5179 [ name = cmr(oldstyle)-it,
5180   load = cmr-it-T1 ]
5181 { encoding = T1,
5182   family   = {hfor,cmor},
5183   shape     = {it,sl} }
5184 {
5185   1 = {250, 50},
5186   2 = {150,-100},
5187   3 = {100,-50},
5188   4 = {150,150},
5189   6 = {200, },
5190   7 = {200, 50},
5191   8 = {150,-50},
5192   9 = {100, 50}
5193 }
5194
5195  $\langle /cmr \rangle$ 
5196  $\langle *pmn \rangle$ 
5197 \SetProtrusion
5198 [ name = pmnx-it,
5199   load = pmnj-it ]
5200 { encoding = OT1,
5201   family   = pmnx,
5202   shape     = {it,sl} }
5203 {
5204   1 = {100,150}
5205 }
5206
5207 \SetProtrusion
5208 [ name = pmnx-it-T1,
5209   load = pmnj-it-T1 ]
5210 { encoding = {T1,LY1},
5211   family   = pmnx,
5212   shape     = {it,sl} }
5213 {
5214   1 = {100,150}
5215 }
5216
5217  $\langle /pmn \rangle$ 
5218  $\langle *ptm \rangle$ 

```

```

5219 \SetProtrusion
5220 [ name      = ptm-it-LY1,
5221   load      = ptm-it-T1 ]
5222 { encoding = {LY1},
5223   family   = {ptm,ptmx,ptmj},
5224   shape     = {it,sl} }
5225 {
5226   -                      = {100,100},
5227   \texttrademark         = {100,100},
5228   \textregistered        = {100,100},
5229   \textcopyright         = {100,100},
5230   \textdegree            = {300,100},
5231   \textminus             = {200,200},
5232   \textellipsis          = {100,200},
5233   \% \texteuro           = { , }, % ?
5234   \textcent              = {100,100},
5235   \textquotesingle       = {500, },
5236   \textflorin            = {100, 70},
5237   \textdagger            = {150,150},
5238   \textdaggerdbl         = {100,100},
5239   \textbullet            = {150,150},
5240   \textonesuperior       = {150,100},
5241   \texttwosuperior       = {150, 50},
5242   \textthreesuperior     = {150, 50},
5243   \textparagraph         = {100, },
5244   \textperiodcentered    = {500,300},
5245   \textonequarter        = { 50, },
5246   \textonehalf           = { 50, },
5247   \textplusminus         = {100,100},
5248   \textmultiply          = {150,150},
5249   \textdivide           = {150,150}
5250 }
5251
5252 </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).

```

5253 <*(b|g|u|m)>
5254 \SetProtrusion
5255 <m-t> [ name      = OT1-sc,
5256 <bch> [ name      = bch-sc,
5257 <cmr> [ name      = cmr-sc-OT1,
5258 <pad> [ name      = pad-sc,
5259 <pmn> [ name      = pmnj-sc,
5260 <ppl> [ name      = ppl-sc,
5261 <ptm> [ name      = ptm-sc,
5262 <m-t>   load      = default ]
5263 <bch>   load      = bch-default ]
5264 <cmr>   load      = cmr-OT1 ]
5265 <pad>   load      = pad-default ]
5266 <pmn>   load      = pmnj-default ]
5267 <ppl>   load      = ppl-default ]
5268 <ptm>   load      = ptm-default ]
5269 <m-t|bch|pad|pmn> { encoding = OT1,
5270 <cmr|ppl|ptm> { encoding = {OT1,OT4},
5271 <bch>   family    = bch,
5272 <cmr>   family    = cmr,
5273 <pad>   family    = {pad,padx,padj},

```

```

5274 <pmn>    family    = pmnj,
5275 <ppl>     family    = {ppl,pplx,pplj},
5276 <ptm>     family    = {ptm,ptmx,ptmj},
5277         shape      = sc }
5278 {
5279     a = {50,50},
5280 <cmr|pad|ppl|ptm> \ae = {50, },
5281 <bch|pmn>      c = {50, },
5282 <bch|pad|pmn>  d = { ,50},
5283 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
5284 <bch|pad|pmn>  g = {50, },
5285 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
5286 <bch>          j = {100, },
5287 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
5288 <ptm>          l = { ,80},
5289 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
5290 <ptm>          013 = { ,80}, % fl
5291 <bch|pad|pmn>   o = {50,50},
5292 <pad|pmn>      \oe = {50, },
5293 <ppl>          p = { 0, 0},
5294 <bch|pad|pmn>   q = {50,70},
5295 <ppl>          q = { 0, },
5296 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5297         t = {50,50},
5298 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5299 <ptm>          y = {80,80}
5300 }
5301
5302 \SetProtrusion
5303 <m-t> [ name      = T1-sc,
5304 <bch> [ name      = bch-sc-T1,
5305 <cmr> [ name      = cmr-sc-T1,
5306 <pad> [ name      = pad-sc-T1,
5307 <pmn> [ name      = pmnj-sc-T1,
5308 <ppl> [ name      = ppl-sc-T1,
5309 <ptm> [ name      = ptm-sc-T1,
5310 <m-t>  load      = T1-default ]
5311 <bch>  load      = bch-T1    ]
5312 <cmr>  load      = cmr-T1    ]
5313 <pad>  load      = pad-T1    ]
5314 <pmn>  load      = pmnj-T1   ]
5315 <ppl>  load      = ppl-T1    ]
5316 <ptm>  load      = ptm-T1    ]
5317 { encoding = {T1,LY1},
5318 <bch>    family    = bch,
5319 <cmr>    family    = cmr,
5320 <pad>    family    = {pad,padx,padj},
5321 <pmn>    family    = pmnj,
5322 <ppl>    family    = {ppl,pplx,pplj},
5323 <ptm>    family    = {ptm,ptmx,ptmj},
5324         shape      = sc }
5325 {
5326     a = {50,50},
5327 <cmr|pad|ppl|ptm> \ae = {50, },
5328 <bch|pmn>      c = {50, },
5329 <bch|pad|pmn>  d = { ,50},
5330 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
5331 <bch|pad|pmn>  g = {50, },
5332 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
5333 <bch>          j = {100, },
5334 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
5335 <ptm>          l = { ,80},
5336 <m-t|bch|cmr|pad|pmn|ppl> 029 = { ,50}, % fl

```



```

5337 <ptm> 029 = { ,80}, % f1
5338 <bch|pad|pmn> o = {50,50},
5339 <bch|pad|pmn> \oe = {50, },
5340 <ppl> p = { 0, 0},
5341 <bch|pad|pmn> q = {50,70},
5342 <ppl> q = { 0, },
5343 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5344 t = {50,50},
5345 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5346 <ptm> y = {80,80}
5347 }
5348
5349 </(big|ugm)>
5350 <*m-t>
5351 \SetProtrusion
5352 [ name = QX-sc,
5353 load = QX-default ]
5354 { encoding = QX,
5355 shape = sc }
5356 {
5357 a = {50,50},
5358 f = { ,50},
5359 j = {50, },
5360 l = { ,50},
5361 013 = { ,50}, % f1
5362 r = { , 0},
5363 t = {50,50},
5364 y = {50,50}
5365 }
5366
5367 </m-t>
5368 <*cmr|bch>
5369 \SetProtrusion
5370 <bch> [ name = bch-sc-T5,
5371 <bch> load = bch-T5 ]
5372 <cmr> [ name = cmr-sc-T5,
5373 <cmr> load = cmr-T5 ]
5374 { encoding = T5,
5375 <bch> family = bch,
5376 <cmr> family = cmr,
5377 shape = sc }
5378 {
5379 a = {50,50},
5380 <bch> c = {50, },
5381 <bch> d = { ,50},
5382 f = { ,50},
5383 <bch> g = {50, },
5384 <bch> j = {100, },
5385 <cmr> j = {50, },
5386 l = { ,50},
5387 <bch> o = {50,50},
5388 <bch> q = { 0, },
5389 <cmr> r = { , 0},
5390 t = {50,50},
5391 y = {50,50}
5392 }
5393
5394 </cmr|bch>
5395 <*pmn>
5396 \SetProtrusion
5397 [ name = pmnx-sc,
5398 load = pmnj-sc ]
5399 { encoding = OT1,

```

```

5400     family    = pmnx,
5401     shape      = sc }
5402   {
5403     1 = {230,180}
5404   }
5405
5406 \SetProtrusion
5407   [ name      = pmnx-sc-T1,
5408     load      = pmnj-sc-T1 ]
5409   { encoding = {T1,LY1},
5410     family    = pmnx,
5411     shape      = sc }
5412   {
5413     1 = {230,180}
5414   }
5415

```

#### 15.8.4 Italic small caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's `fontinstallationguide` suggests `si`.

```

5416 \SetProtrusion
5417   [ name      = pmnj-scit,
5418     load      = pmnj-it  ]
5419   { encoding = OT1,
5420     family    = pmnj,
5421     shape      = {scit,si} }
5422   {
5423     a = {50, },
5424     \ae = { , -50},
5425     b = {20, -50},
5426     c = {50, -50},
5427     d = {20, 0},
5428     e = {20, -50},
5429     f = {10, 0},
5430     012 = {10, -50}, % fi
5431     013 = {10, -50}, % fl
5432     014 = {10, -50}, % ffi
5433     015 = {10, -50}, % ffl
5434     g = {50, -50},
5435     i = {20, -50},
5436     j = {20, 0},
5437     k = {20, },
5438     l = {20, 50},
5439     m = { , -30},
5440     n = { , -30},
5441     o = {50, },
5442     \oe = {50, -50},
5443     p = {20, -50},
5444     q = {50, },
5445     r = {20, 0},
5446     s = {20, -30},
5447     t = {70, },
5448     u = {50, -50},
5449     v = {100, },
5450     w = {100, },
5451     y = {50, },
5452     z = { , -50}
5453   }
5454
5455 \SetProtrusion

```

```

5456 [ name      = pmnj-scit-T1,
5457   load      = pmnj-it-T1 ]
5458 { encoding = {T1,LY1},
5459   family   = pmnj,
5460   shape    = {scit,si} }
5461 {
5462   a = {50, },
5463   \ae = { , -50},
5464   b = {20, -50},
5465   c = {50, -50},
5466   d = {20, 0},
5467   e = {20, -50},
5468   f = {10, 0},
5469   028 = {10, -50}, % fi
5470   029 = {10, -50}, % fl
5471   030 = {10, -50}, % ffi
5472   031 = {10, -50}, % ffl
5473   g = {50, -50},
5474   i = {20, -50},
5475   188 = {20, 0}, % ij
5476   j = {20, 0},
5477   k = {20, },
5478   l = {20, 50},
5479   m = { , -30},
5480   n = { , -30},
5481   o = {50, },
5482   \oe = {50, -50},
5483   p = {20, -50},
5484   q = {50, },
5485   r = {20, 0},
5486   s = {20, -30},
5487   t = {70, },
5488   u = {50, -50},
5489   v = {100, },
5490   w = {100, },
5491   y = {50, },
5492   z = { , -50}
5493 }
5494
5495 \SetProtrusion
5496 [ name      = pmnx-scit,
5497   load      = pmnj-scit ]
5498 { encoding = OT1,
5499   family   = pmnx,
5500   shape    = {scit,si} }
5501 {
5502   1 = {100,150}
5503 }
5504
5505 \SetProtrusion
5506 [ name      = pmnx-scit-T1,
5507   load      = pmnj-scit-T1 ]
5508 { encoding = {T1,LY1},
5509   family   = pmnx,
5510   shape    = {scit,si} }
5511 {
5512   1 = {100,150}
5513 }
5514
5515 </pmn>

```

### 15.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino.

Anybody?

```

5516 \SetProtrusion
5517 <m-t> [ name = textcomp ]
5518 <bch> [ name = bch-textcomp ]
5519 <blg> [ name = blg-textcomp ]
5520 <cmr> [ name = cmr-textcomp ]
5521 <pad> [ name = pad-textcomp ]
5522 <pmn> [ name = pmn-textcomp ]
5523 <ppl> [ name = ppl-textcomp ]
5524 <ptm> [ name = ptm-textcomp ]
5525 <ugm> [ name = ugm-textcomp ]
5526 <m-t> { encoding = TS1 }
5527 <!m-t> { encoding = TS1,
5528 <bch> family = bch }
5529 <blg> family = blg }
5530 <cmr> family = cmr }
5531 <pad> family = {pad,padx,padj} }
5532 <pmn> family = {pmnx,pmnj} }
5533 <ppl> family = {ppl,pplx,pplj} }
5534 <ptm> family = {ptm,ptmx,ptmj} }
5535 <ugm> family = ugm }
5536 {
5537 <blg> \textquotestraightbase = {400,500},
5538 <cmr> \textquotestraightbase = {300,300},
5539 <pad|pmn> \textquotestraightbase = {400,400},
5540 <blg> \textquotestraightdblbase = {300,400},
5541 <cmr|pmn> \textquotestraightdblbase = {300,300},
5542 <pad> \textquotestraightdblbase = {400,400},
5543 <bch|cmr|pad|pmn|ugm> \texttwelveudash = {200,200},
5544 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
5545 <ugm> \textthreequartersemdash = {200,200},
5546 <blg> \textquotesingle = {500,600},
5547 <cmr|pmn> \textquotesingle = {300,400},
5548 <pad> \textquotesingle = {400,500},
5549 <ptm> \textquotesingle = {500,500},
5550 <ugm> \textquotesingle = {300,500},
5551 <bch|cmr|pmn> \textasteriskcentered = {200,300},
5552 <blg> \textasteriskcentered = {150,200},
5553 <pad> \textasteriskcentered = {300,300},
5554 <ugm> \textasteriskcentered = {100,200},
5555 <pmn> \textfactionsolidus = {-200,-200},
5556 <cmr> \textoneoldstyle = {100,100},
5557 <pmn> \textoneoldstyle = { , 50},
5558 <cmr> \textthreeoldstyle = { , 50},
5559 <pad|pmn> \textthreeoldstyle = { 50, },
5560 <cmr> \textfouroldstyle = { 50, 50},
5561 <pad|pmn> \textfouroldstyle = { 50, },
5562 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
5563 <cmr> \textlangle = {400, },
5564 <cmr> \textrangle = { ,400},
5565 <m-t|bch|pmn|ptm> \textminus = {200,200},
5566 <cmr|pad|ppl> \textminus = {300,300},
5567 <blg|ugm> \textminus = {250,300},
5568 <bch|pad|pmn> \textlbrackdbl = {100, },
5569 <blg> \textlbrackdbl = {200, },
5570 <bch|pad|pmn> \textrbrackdbl = { ,100},
5571 <blg> \textrbrackdbl = { ,200},
5572 <pmn> \textasciigrave = {200,500},
5573 <bch|blg|cmr|pad|pmn> \texttildebelow = {200,250},

```

```

5574 <pmn> \textasciibreve = {300,400},
5575 <pmn> \textasciicaron = {300,400},
5576 <pmn> \textacutedbl = {200,300},
5577 <pmn> \textgravedbl = {150,300},
5578 <bch|pmn|ugm> \textdagger = { 80, 80},
5579 <blg> \textdagger = {200,200},
5580 <cmr|pad> \textdagger = {100,100},
5581 <ptm> \textdagger = {150,150},
5582 <blg> \textdaggerdbl = {150,150},
5583 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
5584 <ptm> \textdaggerdbl = {100,100},
5585 <bch> \textbardbl = {100,100},
5586 <blg|ugm> \textbardbl = {150,150},
5587 <bch> \textbullet = {200,200},
5588 <blg> \textbullet = {400,500},
5589 <cmr|pad|pmn> \textbullet = { ,100},
5590 <ptm> \textbullet = {150,150},
5591 <ugm> \textbullet = { 50,100},
5592 <bch|cmr|pmn> \textcelsius = { 50, },
5593 <pad> \textcelsius = { 80, },
5594 <bch> \textflorin = { 50, 50},
5595 <blg> \textflorin = {100,100},
5596 <pad|ugm> \textflorin = { ,100},
5597 <pmn> \textflorin = { 50,100},
5598 <ptm> \textflorin = { 50, 70},
5599 <cmr> \textcolonmonetary = { , 50},
5600 <pad|pmn> \textcolonmonetary = { 50, },
5601 <pmn> \textinterrobang = { ,100},
5602 <pmn> \textinterrobangdown = {100, },
5603 <m-t|pad|ptm> \texttrademark = {100,100},
5604 <bch> \texttrademark = {150,150},
5605 <blg|cmr|ppl> \texttrademark = {200,200},
5606 <pmn> \texttrademark = { 50, 50},
5607 <ugm> \texttrademark = {100,150},
5608 <bch|ugm> \textcent = { 50, },
5609 <ptm> \textcent = {100,100},
5610 <bch> \textsterling = { 50, },
5611 <ugm> \textsterling = { , 50},
5612 <bch> \textbrokenbar = {200,200},
5613 <blg> \textbrokenbar = {250,250},
5614 <ugm> \textbrokenbar = {200,300},
5615 <pmn> \textasciidieresis = {300,400},
5616 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
5617 <pmn> \textcopyright = {100,150},
5618 <ppl> \textcopyright = {200,200},
5619 <bch|cmr|ugm> \textordfeminine = {100,200},
5620 <pad|pmn> \textordfeminine = {200,200},
5621 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
5622 <blg> \textlnot = {200,100},
5623 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
5624 <pmn> \textregistered = { 50,150},
5625 <ppl> \textregistered = {200,200},
5626 <pmn> \textasciimacron = {150,200},
5627 <m-t|ppl|ptm> \textdegree = {300,300},
5628 <bch> \textdegree = {150,200},
5629 <blg|ugm> \textdegree = {200,200},
5630 <cmr|pad> \textdegree = {400,400},
5631 <pmn> \textdegree = {150,400},
5632 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
5633 <blg> \textpm = {100,100},
5634 <ptm> \textpm = { 50, 80},
5635 <bch|blg|ugm> \texttwosuperior = {100,200},
5636 <cmr> \texttwosuperior = { 50,100},

```

```

5637 <pad|pmn> \texttwosuperior = {200,200},
5638 <ptm> \texttwosuperior = { 50, 50},
5639 <bch|blg|ugm> \textthreesuperior = {100,200},
5640 <cmr> \textthreesuperior = { 50,100},
5641 <pad|pmn> \textthreesuperior = {200,200},
5642 <ptm> \textthreesuperior = { 50, 50},
5643 <pmn> \textasciicute = {300,400},
5644 <bch|ugm> \textmu = { ,100},
5645 <bch|pad|pmn> \textparagraph = { ,100},
5646 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
5647 <blg> \textperiodcentered = {400,500},
5648 <ptm> \textperiodcentered = {300,300},
5649 <ugm> \textperiodcentered = {200,500},
5650 <bch|blg|ugm> \textonesuperior = {200,300},
5651 <cmr|pad|pmn> \textonesuperior = {200,200},
5652 <ptm> \textonesuperior = {100,100},
5653 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
5654 <blg|cmr> \textordmasculine = {100,200},
5655 <bch|cmr|pmn> \texteuro = {100, },
5656 <pad> \texteuro = { 50,100},
5657 <bch> \texttimes = {200,200},
5658 <blg|ptm> \texttimes = {100,100},
5659 <cmr> \texttimes = {150,250},
5660 <pad> \texttimes = {100,150},
5661 <pmn> \texttimes = { 70,100},
5662 <ugm> \texttimes = {200,300},
5663 <bch|pad|pmn> \textdiv = {150,200}
5664 <blg> \textdiv = {100,100}
5665 <cmr> \textdiv = {150,250}
5666 <ptm> \textdiv = { 50,100},
5667 <ugm> \textdiv = {200,300},
5668 <ptm> \textperthousand = { ,50}
5669 <ugm> \textsection = { ,100},
5670 <ugm> \textonehalf = { 50,100},
5671 <ugm> \textonequarter = { 50,100},
5672 <ugm> \textthreequarters = { 50,100},
5673 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

5674 }
5675
5676 <*cmr|pad|pmn|ugm>
5677 \SetProtrusion
5678 <cmr> [ name = cmr-textcomp-it ]
5679 <pad> [ name = pad-textcomp-it ]
5680 <pmn> [ name = pmn-textcomp-it ]
5681 <ugm> [ name = ugm-textcomp-it ]
5682 { encoding = TS1,
5683 <cmr> family = cmr,
5684 <pad> family = {pad,padx,padj},
5685 <pmn> family = {pmnx,pmnj},
5686 <ugm> family = ugm,
5687 <!ugm> shape = {it,sl} }
5688 <ugm> shape = it }
5689 {
5690 <cmr> \textquotestraightbase = {300,600},
5691 <pad|pmn> \textquotestraightbase = {400,400},
5692 <cmr> \textquotestraightdblbase = {300,600},
5693 <pad> \textquotestraightdblbase = {300,400},
5694 <pmn> \textquotestraightdblbase = {300,300},
5695 \texttwelveudash = {200,200},
5696 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
5697 <ugm> \textthreequartersemdash = {200,200},

```

5698	<i>cmr</i>	<code>\textquotesingle</code>	= {600,300},
5699	<i>pad</i>	<code>\textquotesingle</code>	= {800,100},
5700	<i>pmn</i>	<code>\textquotesingle</code>	= {300,200},
5701	<i>ugm</i>	<code>\textquotesingle</code>	= {500,500},
5702	<i>cmr</i>	<code>\textasteriskcentered</code>	= {300,200},
5703	<i>pad</i>	<code>\textasteriskcentered</code>	= {500,100},
5704	<i>pmn</i>	<code>\textasteriskcentered</code>	= {200,300},
5705	<i>ugm</i>	<code>\textasteriskcentered</code>	= {300,150},
5706	<i>pmn</i>	<code>\textfractionsolidus</code>	= {-200,-200},
5707	<i>cmr</i>	<code>\textoneoldstyle</code>	= {100, 50},
5708	<i>pad</i>	<code>\textoneoldstyle</code>	= {100, },
5709	<i>pmn</i>	<code>\textoneoldstyle</code>	= { 50, },
5710	<i>pad</i>	<code>\texttwooldstyle</code>	= { 50, },
5711	<i>pmn</i>	<code>\texttwooldstyle</code>	= {-50, },
5712	<i>cmr</i>	<code>\textthreeoldstyle</code>	= {100, 50},
5713	<i>pmn</i>	<code>\textthreeoldstyle</code>	= {-100, },
5714	<i>cmr</i>	<code>\textfouroldstyle</code>	= { 50, 50},
5715	<i>pad</i>	<code>\textfouroldstyle</code>	= { 50,100},
5716	<i>cmr</i>	<code>\textsevenoldstyle</code>	= { 50, 80},
5717	<i>pad</i>	<code>\textsevenoldstyle</code>	= { 50, },
5718	<i>pmn</i>	<code>\textsevenoldstyle</code>	= { 20, },
5719	<i>cmr</i>	<code>\textlangle</code>	= {400, },
5720	<i>cmr</i>	<code>\textrangle</code>	= { ,400},
5721	<i>cmr pad</i>	<code>\textminus</code>	= {300,300},
5722	<i>pmn</i>	<code>\textminus</code>	= {200,200},
5723	<i>ugm</i>	<code>\textminus</code>	= {250,300},
5724	<i>pad pmn</i>	<code>\textlbrackdbl</code>	= {100, },
5725	<i>pad pmn</i>	<code>\textrbrackdbl</code>	= { ,100},
5726	<i>pmn</i>	<code>\textasciigrave</code>	= {300,300},
5727	<i>cmr pad pmn</i>	<code>\texttildelow</code>	= {200,250},
5728	<i>pmn</i>	<code>\textasciibreve</code>	= {300,300},
5729	<i>pmn</i>	<code>\textasciicaron</code>	= {300,300},
5730	<i>pmn</i>	<code>\textacutedbl</code>	= {200,300},
5731	<i>pmn</i>	<code>\textgravedbl</code>	= {150,300},
5732	<i>cmr</i>	<code>\textdagger</code>	= {100,100},
5733	<i>pad</i>	<code>\textdagger</code>	= {200,100},
5734	<i>pmn</i>	<code>\textdagger</code>	= { 80, 50},
5735	<i>ugm</i>	<code>\textdagger</code>	= { 80, 80},
5736	<i>cmr pad</i>	<code>\textdaggerdbl</code>	= { 80, 80},
5737	<i>pmn</i>	<code>\textdaggerdbl</code>	= { 80, 50},
5738	<i>ugm</i>	<code>\textbardbl</code>	= {150,150},
5739	<i>cmr</i>	<code>\textbullet</code>	= {200,100},
5740	<i>pad</i>	<code>\textbullet</code>	= {300, },
5741	<i>pmn</i>	<code>\textbullet</code>	= { 30, 70},
5742	<i>ugm</i>	<code>\textbullet</code>	= { 50,100},
5743	<i>cmr</i>	<code>\textcelsius</code>	= {100, },
5744	<i>pad</i>	<code>\textcelsius</code>	= {200, },
5745	<i>pmn</i>	<code>\textcelsius</code>	= { 50,-50},
5746	<i>pad</i>	<code>\textflorin</code>	= {100, },
5747	<i>pmn</i>	<code>\textflorin</code>	= { 50,100},
5748	<i>ugm</i>	<code>\textflorin</code>	= { ,100},
5749	<i>cmr</i>	<code>\textcolonmonetary</code>	= {150, },
5750	<i>pad</i>	<code>\textcolonmonetary</code>	= {100, },
5751	<i>pmn</i>	<code>\textcolonmonetary</code>	= { 50,-50},
5752	<i>cmr pad</i>	<code>\texttrademark</code>	= {200, },
5753	<i>pmn</i>	<code>\texttrademark</code>	= { 50,100},
5754	<i>ugm</i>	<code>\texttrademark</code>	= {150, 50},
5755	<i>ugm</i>	<code>\textcent</code>	= { 50, },
5756	<i>ugm</i>	<code>\textsterling</code>	= { , 50},
5757	<i>ugm</i>	<code>\textbrokenbar</code>	= {200,300},
5758	<i>pmn</i>	<code>\textasciidieresis</code>	= {300,200},
5759	<i>cmr</i>	<code>\textcopyright</code>	= {100, },
5760	<i>pad</i>	<code>\textcopyright</code>	= {200,100},

```

5761 <pmn> \textcopyright = {100,150},
5762 <ugm> \textcopyright = {300, },
5763 <cmr> \textordfeminine = {100,100},
5764 <pmn> \textordfeminine = {200,200},
5765 <ugm> \textordfeminine = {100,200},
5766 <cmr|pad> \textlnot = {300, },
5767 <pmn|ugm> \textlnot = {200, },
5768 <cmr> \textregistered = {100, },
5769 <pad> \textregistered = {200,100},
5770 <pmn> \textregistered = { 50,150},
5771 <ugm> \textregistered = {300, },
5772 <pmn> \textasciimacron = {150,200},
5773 <cmr|pad> \textdegree = {500,100},
5774 <pmn> \textdegree = {150,150},
5775 <ugm> \textdegree = {300,200},
5776 <cmr> \textpm = {150,100},
5777 <pad> \textpm = {200,150},
5778 <pmn|ugm> \textpm = {150,200},
5779 <cmr> \textonesuperior = {400, },
5780 <pad> \textonesuperior = {300,100},
5781 <pmn> \textonesuperior = {200,100},
5782 <ugm> \textonesuperior = {300,300},
5783 <cmr> \texttwosuperior = {400, },
5784 <pad> \texttwosuperior = {300, },
5785 <pmn> \texttwosuperior = {200,100},
5786 <ugm> \texttwosuperior = {300,200},
5787 <cmr> \textthreesuperior = {400, },
5788 <pad> \textthreesuperior = {300, },
5789 <pmn> \textthreesuperior = {200,100},
5790 <ugm> \textthreesuperior = {300,200},
5791 <ugm> \textmu = { ,100},
5792 <pmn> \textasciiaacute = {300,200},
5793 <cmr> \textparagraph = {200, },
5794 <pmn> \textparagraph = { ,100},
5795 <cmr> \textperiodcentered = {500,500},
5796 <pad|pmn|ugm> \textperiodcentered = {300,400},
5797 <cmr> \textordmasculine = {100,100},
5798 <pmn> \textordmasculine = {200,200},
5799 <ugm> \textordmasculine = {300,200},
5800 <cmr> \texteuro = {200, },
5801 <pad> \texteuro = {100, },
5802 <pmn> \texteuro = {100,-50},
5803 <cmr> \texttimes = {200,200},
5804 <pad> \texttimes = {200,100},
5805 <pmn> \texttimes = { 70,100},
5806 <ugm> \texttimes = {200,300},
5807 <cmr|pad> \textdiv = {200,200},
5808 <pmn> \textdiv = {150,200},
5809 <ugm> \textdiv = {200,300},
5810 <ugm> \textsection = { ,200},
5811 <ugm> \textonehalf = { 50,100},
5812 <ugm> \textonequarter = { 50,100},
5813 <ugm> \textthreequarters = { 50,100},
5814 <ugm> \textsurd = { ,100}
5815 }
5816
5817 </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}
```

```
5818 (*cmr*)
5819 \SetProtrusion
5820 [ name = cmr-math-letters ]
5821 { encoding = OML,
5822   family = cmm,
5823   series = {m,b},
5824   shape = it }
5825 {
5826   A = {100, 50}, % \mathnormal
5827   B = { 50,   },
5828   C = { 50,   },
5829   D = { 50, 50},
5830   E = { 50,   },
5831   F = {100, 50},
5832   G = { 50, 50},
5833   H = { 50, 50},
5834   I = { 50, 50},
5835   J = {150, 50},
5836   K = { 50,100},
5837   L = { 50, 50},
5838   M = { 50,   },
5839   N = { 50,   },
5840   O = { 50,   },
5841   P = { 50,   },
5842   Q = { 50, 50},
5843   R = { 50,   },
5844   S = { 50,   },
5845   T = { 50,100},
5846   U = { 50, 50},
5847   V = {100,100},
5848   W = { 50,100},
5849   X = { 50,100},
5850   Y = {100,100},
5851   f = {100,100},
5852   h = {   ,100},
5853   i = {   , 50},
5854   j = {   , 50},
5855   k = {   , 50},
5856   r = {   , 50},
5857   v = {   , 50},
5858   w = {   , 50},
5859   x = {   , 50},
5860 "OB = { 50,100}, % \alpha
```

```

5861 "0C = { 50, 50}, % \beta
5862 "0D = {200,150}, % \gamma
5863 "0E = { 50, 50}, % \delta
5864 "0F = { 50, 50}, % \epsilon
5865 "10 = { 50,150}, % \zeta
5866 "12 = { 50, }, % \theta
5867 "13 = { ,100}, % \iota
5868 "14 = { ,100}, % \kappa
5869 "15 = {100, 50}, % \lambda
5870 "16 = { , 50}, % \mu
5871 "17 = { , 50}, % \nu
5872 "18 = { , 50}, % \xi
5873 "19 = { 50,100}, % \pi
5874 "1A = { 50, 50}, % \rho
5875 "1B = { ,150}, % \sigma
5876 "1C = { 50,150}, % \tau
5877 "1D = { 50, 50}, % \upsilon
5878 "1F = { 50,100}, % \chi
5879 "20 = { 50, 50}, % \psi
5880 "21 = { , 50}, % \omega
5881 "22 = { , 50}, % \varepsilon
5882 "23 = { , 50}, % \vartheta
5883 "24 = { , 50}, % \varpi
5884 "25 = {100, }, % \varrho
5885 "26 = {100,100}, % \varsigma
5886 "27 = { 50, 50}, % \varphi
5887 "28 = {100,100}, % \leftharpoonup
5888 "29 = {100,100}, % \leftharpoondown
5889 "2A = {100,100}, % \rightharpoonup
5890 "2B = {100,100}, % \rightharpoondown
5891 "2C = {300,200}, % \lhook
5892 "2D = {200,300}, % \rhook
5893 "2E = { ,100}, % \triangleright
5894 "2F = {100, }, % \triangleleft
5895 "3A = { ,500}, % ., \ldotp
5896 "3B = { ,500}, % ,
5897 "3C = {200,100}, % <
5898 "3D = {300,400}, % /
5899 "3E = {100,200}, % >
5900 "3F = {200,200}, % \star
5901 "5B = { ,100}, % \flat
5902 "5E = {200,200}, % \smile
5903 "5F = {200,200}, % \frown
5904 "7C = {100, }, % \jmath
5905 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

5906 }
5907

```

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}

```

```

5908 \SetProtrusion
5909 [ name = cmr-math-symbols ]
5910 { encoding = OMS,
5911   family = cmsy,
5912   series = {m,b},
5913   shape = n }
5914 {
5915   A = {150, 50}, % \mathcal

```

```

5916     C = {    ,100},
5917     D = {    , 50},
5918     F = { 50,150},
5919     I = {    ,100},
5920     J = {100,150},
5921     K = {    ,100},
5922     L = {100,   },
5923     M = { 50, 50},
5924     N = { 50,100},
5925     P = {    , 50},
5926     Q = { 50,   },
5927     R = {    , 50},
5928     T = { 50,150},
5929     V = { 50, 50},
5930     W = {    , 50},
5931     X = {100,100},
5932     Y = {100,   },
5933     Z = {100,150},
5934     "00 = {300,300}, % -
5935     "01 = {    ,700}, % \cdot, \cdotp
5936     "02 = {150,250}, % \times
5937     "03 = {150,250}, % *, \ast
5938     "04 = {200,300}, % \div
5939     "05 = {150,250}, % \diamond
5940     "06 = {200,200}, % \pm
5941     "07 = {200,200}, % \mp
5942     "08 = {100,100}, % \oplus
5943     "09 = {100,100}, % \ominus
5944     "0A = {100,100}, % \otimes
5945     "0B = {100,100}, % \oslash
5946     "0C = {100,100}, % \odot
5947     "0D = {100,100}, % \bigcirc
5948     "0E = {100,100}, % \circ
5949     "0F = {100,100}, % \bullet
5950     "10 = {100,100}, % \asymp
5951     "11 = {100,100}, % \equiv
5952     "12 = {200,100}, % \subseteq
5953     "13 = {100,200}, % \supseteq
5954     "14 = {200,100}, % \leq
5955     "15 = {100,200}, % \geq
5956     "16 = {200,100}, % \preceq
5957     "17 = {100,200}, % \succeq
5958     "18 = {200,200}, % \sim
5959     "19 = {150,150}, % \approx
5960     "1A = {200,100}, % \subset
5961     "1B = {100,200}, % \supset
5962     "1C = {200,100}, % \ll
5963     "1D = {100,200}, % \gg
5964     "1E = {300,100}, % \prec
5965     "1F = {100,300}, % \succ
5966     "20 = {100,200}, % \leftarrow
5967     "21 = {200,100}, % \rightarrow
5968     "22 = {100,100}, % \uparrow
5969     "23 = {100,100}, % \downarrow
5970     "24 = {100,100}, % \leftrightarrows
5971     "25 = {100,100}, % \nearrow
5972     "26 = {100,100}, % \searrow
5973     "27 = {100,100}, % \simeq
5974     "28 = {100,100}, % \Leftarrow
5975     "29 = {100,100}, % \Rightarrow
5976     "2A = {100,100}, % \Uparrow
5977     "2B = {100,100}, % \Downarrow
5978     "2C = {100,100}, % \Leftrightarrow

```

```

5979 "2D = {100,100}, % \narrow
5980 "2E = {100,100}, % \swarrow
5981 "2F = { ,100}, % \propto
5982 "30 = { ,400}, % \prime
5983 "31 = {100,100}, % \infty
5984 "32 = {150,100}, % \in
5985 "33 = {100,150}, % \ni
5986 "34 = {100,100}, % \triangle, \bigtriangleup
5987 "35 = {100,100}, % \bigtriangledown
5988 "38 = { ,100}, % \forall
5989 "39 = {100, }, % \exists
5990 "3A = {200, }, % \neg
5991 "3E = {200,200}, % \top
5992 "3F = {200,200}, % \bot, \perp
5993 "5E = {100,200}, % \wedge
5994 "5F = {100,200}, % \vee
5995 "60 = { ,300}, % \vdash
5996 "61 = {300, }, % \dashv
5997 "62 = {100,100}, % \lfloor
5998 "63 = {100,100}, % \rfloor
5999 "64 = {100,100}, % \lceil
6000 "65 = {100,100}, % \rceil
6001 "66 = {150, }, % \lbrace
6002 "67 = { ,150}, % \rbrace
6003 "68 = {400, }, % \langle
6004 "69 = { ,400}, % \rangle
6005 "6C = {100,100}, % \updownarrow
6006 "6D = {100,100}, % \Updownarrow
6007 "6E = {100,300}, % \, \backslash, \setminus
6008 "72 = {100,100}, % \nabla
6009 "79 = {200,200}, % \dagger
6010 "7A = {100,100}, % \ddagger
6011 "7B = {100, }, % \mathparagraph
6012 "7C = {100,100}, % \clubsuit
6013 "7D = {100,100}, % \diamondsuit
6014 "7E = {100,100}, % \heartsuit
6015 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

6016 }
6017

```

We don't bother about 'largesymbols', since it will only be used in display math, where protrusion doesn't work anyway. It's declared as:

```
\DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}
```

```

6018 </cmr>
6019 </cfg-t>

```

### 15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
6020 <*cfg-u>
```

Symbol font 'a'.

```

6021 <*msa>
6022 \SetProtrusion
6023 [ name = AMS-a ]
6024 { encoding = U,
6025   family = msa }

```

```

6026 {
6027   "05 = {150,250}, % \centerdot
6028   "06 = {100,100}, % \lozenge
6029   "07 = { 50, 50}, % \blacklozenge
6030   "08 = { 50, 50}, % \circlearrowright
6031   "09 = { 50, 50}, % \circlearrowleft
6032   "0A = {100,100}, % \rightleftharpoons
6033   "0B = {100,100}, % \leftrightharpoons
6034   "0D = {-50,200}, % \Vdash
6035   "0E = {-50,200}, % \Vvdash
6036   "0F = {-70,150}, % \vDash
6037   "10 = {100,150}, % \twoheadrightarrow
6038   "11 = {100,150}, % \twoheadleftarrow
6039   "12 = { 50,100}, % \leftleftarrows
6040   "13 = { 50, 80}, % \rightrightarrows
6041   "14 = {120,120}, % \upuparrows
6042   "15 = {120,120}, % \downdownarrows
6043   "16 = {200,200}, % \upharpoonright
6044   "17 = {200,200}, % \downharpoonright
6045   "18 = {200,200}, % \upharpoonleft
6046   "19 = {200,200}, % \downharpoonleft
6047   "1A = { 80,100}, % \rightarrowtail
6048   "1B = { 80,100}, % \leftarrowtail
6049   "1C = { 50, 50}, % \leftrightarrows
6050   "1D = { 50, 50}, % \rightleftarrows
6051   "1E = {250,   }, % \Lsh
6052   "1F = {   ,250}, % \Rsh
6053   "20 = {100,100}, % \rightsquigarrow
6054   "21 = {100,100}, % \leftrightsquigarrow
6055   "22 = {100, 50}, % \looparrowleft
6056   "23 = { 50,100}, % \looparrowright
6057   "24 = { 50, 80}, % \circeq
6058   "25 = {   ,100}, % \succsim
6059   "26 = {   ,100}, % \gtrsim
6060   "27 = {   ,100}, % \gtrapprox
6061   "28 = {150, 50}, % \multimap
6062   "2B = {100,150}, % \doteqdot
6063   "2C = {100,150}, % \triangleq
6064   "2D = {100, 50}, % \precsim
6065   "2E = {100, 50}, % \lesssim
6066   "2F = { 50, 50}, % \lessapprox
6067   "30 = {100, 50}, % \eqslantless
6068   "31 = { 50, 50}, % \eqslantgtr
6069   "32 = {100, 50}, % \curlyeqprec
6070   "33 = { 50,100}, % \curlyeqsucc
6071   "34 = {100, 50}, % \preccurlyeq
6072   "36 = { 50,   }, % \leqslant
6073   "38 = {   , 50}, % \backprime
6074   "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
6075   "3C = { 50,100}, % \succcurlyeq
6076   "3E = {   , 50}, % \geqslant
6077   "40 = {   , 50}, % \sqsubset
6078   "41 = { 50,   }, % \sqsupset
6079   "42 = {   ,150}, % \vartriangleright, \rhd
6080   "43 = {150,   }, % \vartriangleleft, \lhd
6081   "44 = {   ,100}, % \trianglerighteq, \unrhd
6082   "45 = {100,   }, % \trianglelefteq, \unlhd
6083   "46 = {100,100}, % \bigstar
6084   "48 = { 50, 50}, % \blacktriangledown
6085   "49 = {   ,100}, % \blacktriangleright
6086   "4A = {100,   }, % \blacktriangleleft
6087   "4B = {   ,150}, % \dashrightarrow (the arrow)
6088   "4C = {150,   }, % \dashleftarrow

```

```

6089 "4D = { 50, 50}, % \vartriangle
6090 "4E = { 50, 50}, % \blacktriangle
6091 "4F = { 50, 50}, % \triangledown
6092 "50 = { 50, 50}, % \eqcirc
6093 "56 = { ,150}, % \Rrightarrow
6094 "57 = {150, }, % \Lleftarrow
6095 "58 = {100,300}, % \checkmark
6096 "5C = { 50, 50}, % \angle
6097 "5D = { 50, 50}, % \measuredangle
6098 "5E = { 50, 50}, % \sphericalangle
6099 "5F = { , 50}, % \varpropto
6100 "60 = {100,100}, % \smallsmile
6101 "61 = {100,100}, % \smallfrown
6102 "62 = { 50, }, % \Subset
6103 "63 = { , 50}, % \Supset
6104 "66 = {150,150}, % \curlywedge
6105 "67 = {150,150}, % \curlyvee
6106 "68 = { 50,150}, % \leftthreetimes
6107 "69 = {100, 50}, % \rightthreetimes
6108 "6C = { 50, 50}, % \bumpeq
6109 "6D = { 50, 50}, % \Bumpeq
6110 "6E = {100, }, % \lll
6111 "6F = { ,100}, % \ggg
6112 "70 = { 50,100}, % \ulcorner
6113 "71 = {100, 50}, % \urcorner
6114 "75 = {150,200}, % \dotplus
6115 "76 = { 50,100}, % \backsim
6116 "78 = { 50,100}, % \llcorner
6117 "79 = {100, 50}, % \lrcorner
6118 "7C = {100,100}, % \intercal
6119 "7D = { 50, 50}, % \circledcirc
6120 "7E = { 50, 50}, % \circledast
6121 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

6122 }
6123
6124 </msa>

```

Symbol font 'b'.

```

6125 <*msb>
6126 \SetProtrusion
6127 [ name = AMS-b ]
6128 { encoding = U,
6129   family = msb }
6130 {
6131   A = { 50, 50}, % \mathbb
6132   C = { 50, 50},
6133   G = { , 50},
6134   L = { , 50},
6135   P = { , 50},
6136   R = { , 50},
6137   T = { , 50},
6138   V = { 50, 50},
6139   X = { 50, 50},
6140   Y = { 50, 50},
6141   "00 = { 50, 50}, % \lvertneqq
6142   "01 = { 50, 50}, % \gvertneqq
6143   "02 = { 50, 50}, % \nleq
6144   "03 = { 50, 50}, % \ngeq
6145   "04 = {100, 50}, % \nless
6146   "05 = { 50,150}, % \ngtr
6147   "06 = {100, 50}, % \nprec

```

```

6148 "07 = { 50,150}, % \nsucc
6149 "08 = { 50, 50}, % \lneqq
6150 "09 = { 50, 50}, % \gneqq
6151 "0A = {100,100}, % \lneqslant
6152 "0B = {100,100}, % \ngeqslant
6153 "0C = {100, 50}, % \lneq
6154 "0D = { 50,100}, % \gneq
6155 "0E = {100, 50}, % \npreceq
6156 "0F = { 50,100}, % \nsucceq
6157 "10 = { 50,  }, % \precnsim
6158 "11 = { 50, 50}, % \succnsim
6159 "12 = { 50, 50}, % \lnsim
6160 "13 = { 50, 50}, % \gnsim
6161 "14 = { 50, 50}, % \lneqq
6162 "15 = { 50, 50}, % \ngeqq
6163 "16 = { 50, 50}, % \precneqq
6164 "17 = { 50, 50}, % \succneqq
6165 "18 = { 50, 50}, % \precnapprox
6166 "19 = { 50, 50}, % \succnapprox
6167 "1A = { 50, 50}, % \lnapprox
6168 "1B = { 50, 50}, % \gnapprox
6169 "1C = {150,200}, % \nsim
6170 "1D = { 50, 50}, % \ncong
6171 "1E = {100,150}, % \diagup
6172 "1F = {100,150}, % \diagdown
6173 "20 = {100, 50}, % \varsubsetneq
6174 "21 = { 50,100}, % \varsupsetneq
6175 "22 = {100, 50}, % \subsetneqq
6176 "23 = { 50,100}, % \supsetneqq
6177 "24 = {100, 50}, % \subsetneqq
6178 "25 = { 50,100}, % \supsetneqq
6179 "26 = {100, 50}, % \varsubsetneqq
6180 "27 = { 50,100}, % \varsupsetneqq
6181 "28 = {100, 50}, % \subsetneq
6182 "29 = { 50,100}, % \supsetneq
6183 "2A = {100, 50}, % \subseteq
6184 "2B = { 50,100}, % \supseteq
6185 "2C = { 50,100}, % \nparallel
6186 "2D = {100,150}, % \nmid
6187 "2E = {150,150}, % \nshortmid
6188 "2F = {100,100}, % \nshortparallel
6189 "30 = {  ,150}, % \nvdash
6190 "31 = {  ,150}, % \nVdash
6191 "32 = {  ,100}, % \nvDash
6192 "33 = {  ,100}, % \nVDash
6193 "34 = {  ,100}, % \ntrianglerighteq
6194 "35 = {100,  }, % \trianglelefteq
6195 "36 = {100,  }, % \triangleleft
6196 "37 = {  ,100}, % \triangleright
6197 "38 = {100,200}, % \leftarrow
6198 "39 = {100,200}, % \rightarrow
6199 "3A = {100,100}, % \Leftarrow
6200 "3B = { 50,100}, % \Rightarrow
6201 "3C = {100,100}, % \Leftrightarrow
6202 "3D = {100,200}, % \leftrightharrow
6203 "3E = { 50, 50}, % \divideontimes
6204 "3F = { 50, 50}, % \varnothing
6205 "60 = {200,  }, % \Finv
6206 "61 = {  , 50}, % \Game
6207 "68 = {100,100}, % \eqsim
6208 "69 = { 50,  }, % \beth
6209 "6A = { 50,  }, % \gimel
6210 "6B = {150,  }, % \daleth

```

```

6211 "6C = {200, }, % \lessdot
6212 "6D = { ,200}, % \gtrdot
6213 "6E = {100,200}, % \ltimes
6214 "6F = {150,100}, % \rtimes
6215 "70 = { 50,100}, % \shortmid
6216 "71 = { 50, 50}, % \shortparallel
6217 "72 = {200,300}, % \smallsetminus
6218 "73 = {100,200}, % \thicksim
6219 "74 = { 50,100}, % \thickapprox
6220 "75 = { 50, 50}, % \approxeq
6221 "76 = { 50,100}, % \succapprox
6222 "77 = { 50, 50}, % \precapprox
6223 "78 = {100,100}, % \curvearrowleft
6224 "79 = { 50,150}, % \curvearrowright
6225 "7A = { 50,200}, % \digamma
6226 "7B = {100, 50}, % \varkappa
6227 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

6228 }
6229
6230 (/msb)

```

### 15.8.8 Euler

Euler Roman font (package `euler`).

```

6231 (*eur)
6232 \SetProtrusion
6233 [ name = euler ]
6234 { encoding = U,
6235   family = eur }
6236 {
6237   "01 = {100,100},
6238   "03 = {100,150},
6239   "06 = { ,100},
6240   "07 = {100,150},
6241   "08 = {100,100},
6242   "0A = {100,100},
6243   "0B = { , 50},
6244   "0C = { ,100},
6245   "0D = {100,100},
6246   "0E = { ,100},
6247   "0F = {100,100},
6248   "10 = {100,100},
6249   "13 = { ,100},
6250   "14 = { ,100},
6251   "15 = { , 50},
6252   "16 = { , 50},
6253   "17 = { 50,100},
6254   "18 = { 50,100},
6255   "1A = { , 50},
6256   "1B = { , 50},
6257   "1C = { 50,100},
6258   "1D = { 50,100},
6259   "1E = { 50,100},
6260   "1F = { 50,100},
6261   "20 = { , 50},
6262   "21 = { , 50},
6263   "22 = { 50,100},
6264   "24 = { , 50},
6265   "27 = { 50,100},

```



```

6266      1 = {100,100},
6267      7 = { 50,100},
6268     "3A = {300,500},
6269     "3B = {200,400},
6270     "3C = {200,100},
6271     "3D = {200,200},
6272     "3E = {100,200},
6273      A = {  ,100},
6274      D = {  , 50},
6275      J = { 50,  },
6276      K = {  , 50},
6277      L = {  , 50},
6278      Q = {  , 50},
6279      T = { 50,  },
6280      X = { 50, 50},
6281      Y = { 50,  },
6282      h = {  , 50},
6283      k = {  , 50}
6284  }
6285

```

Extended by the `eulervm` package.

```

6286 \SetProtrusion
6287 [ name = euler-vm,
6288   load = euler ]
6289 { encoding = U,
6290   family = zeur }
6291 {
6292   "28 = {100,200},
6293   "29 = {100,200},
6294   "2A = {100,150},
6295   "2B = {100,150},
6296   "2C = {200,300},
6297   "2D = {200,300},
6298   "2E = {  ,100},
6299   "2F = {100,  },
6300   "3F = {150,150},
6301   "5B = {  ,100},
6302   "5E = {100,100},
6303   "5F = {100,100},
6304   "80 = {  , 50},
6305   "81 = {200,250},
6306   "82 = {100,200}
6307 }
6308
6309 /eur

```

Euler Script font (`euca1`).

```

6310 *eus
6311 \SetProtrusion
6312 [ name = euscript ]
6313 { encoding = U,
6314   family = eus }
6315 {
6316   A = {100,100},
6317   B = { 50,100},
6318   C = { 50, 50},
6319   D = { 50,100},
6320   E = { 50,100},
6321   F = { 50,  },
6322   G = { 50,  },
6323   H = {  ,100},
6324   K = {  , 50},

```

```

6325     L = { ,150},
6326     M = { , 50},
6327     N = { , 50},
6328     O = { 50, 50},
6329     P = { 50, 50},
6330     T = { ,100},
6331     U = { , 50},
6332     V = { 50, 50},
6333     W = { 50, 50},
6334     X = { 50, 50},
6335     Y = { 50, },
6336     Z = { 50,100},
6337     "00 = {250,250},
6338     "18 = {200,200},
6339     "3A = {200,150},
6340     "40 = { ,100},
6341     "5E = {100,100},
6342     "5F = {100,100},
6343     "66 = { 50, },
6344     "67 = { , 50},
6345     "6E = {200,200}
6346 }
6347
6348 \SetProtrusion
6349 [ name = euscript-vm,
6350   load = euscript ]
6351 { encoding = U,
6352   family = zeus }
6353 {
6354   "01 = {600,600},
6355   "02 = {200,200},
6356   "03 = {200,200},
6357   "04 = {200,200},
6358   "05 = {150,150},
6359   "06 = {200,200},
6360   "07 = {200,200},
6361   "08 = {100,100},
6362   "09 = {100,100},
6363   "0A = {100,100},
6364   "0B = {100,100},
6365   "0C = {100,100},
6366   "0D = {100,100},
6367   "0E = {150,150},
6368   "0F = {100,100},
6369   "10 = {150,150},
6370   "11 = {100,100},
6371   "12 = {150,100},
6372   "13 = {100,150},
6373   "14 = {150,100},
6374   "15 = {100,150},
6375   "16 = {200,100},
6376   "17 = {100,200},
6377   "19 = {150,150},
6378   "1A = {150,100},
6379   "1B = {100,150},
6380   "1C = {100,100},
6381   "1D = {100,100},
6382   "1E = {250,100},
6383   "1F = {100,250},
6384   "20 = {150,200},
6385   "21 = {150,200},
6386   "22 = {150,150},
6387   "23 = {150,150},

```

```

6388 "24 = {100,200},
6389 "25 = {150,150},
6390 "26 = {150,150},
6391 "27 = {100,100},
6392 "28 = {100,100},
6393 "29 = {100,150},
6394 "2A = {100,100},
6395 "2B = {100,100},
6396 "2C = {100,100},
6397 "2D = {150,150},
6398 "2E = {150,150},
6399 "2F = {100,100},
6400 "30 = {100,100},
6401 "31 = {100,100},
6402 "32 = {100,100},
6403 "33 = {100,100},
6404 "34 = {100,100},
6405 "35 = {100,100},
6406 "3E = {150,150},
6407 "3F = {150,150},
6408 "60 = { ,200},
6409 "61 = {200, },
6410 "62 = {100,100},
6411 "63 = {100,100},
6412 "64 = {100,100},
6413 "65 = {100,100},
6414 "68 = {300, },
6415 "69 = { ,300},
6416 "6C = {100,100},
6417 "6D = {100,100},
6418 "6F = {100,100},
6419 "72 = {100,100},
6420 "73 = {200,100},
6421 "76 = { ,100},
6422 "77 = {100, },
6423 "78 = { 50, 50},
6424 "79 = {100,100},
6425 "7A = {100,100},
6426 "7D = {150,150},
6427 "7E = {100,100},
6428 "A8 = {100,100},
6429 "A9 = {100,100},
6430 "AB = {200,200},
6431 "BA = { ,200},
6432 "BB = { ,200},
6433 "BD = {200,200},
6434 "DE = {200,200}
6435 }
6436
6437 </eus>

```

Euler Fraktur font (eufrak).

```

6438 <*euf>
6439 \SetProtrusion
6440 [ name = mathfrak ]
6441 { encoding = U,
6442   family = euf }
6443 {
6444   A = { , 50},
6445   B = { , 50},
6446   C = { 50, 50},
6447   D = { , 80},
6448   E = { 50, },

```

```

6449      G = {      , 50},
6450      L = {      , 80},
6451      O = {      , 50},
6452      T = {      , 80},
6453      X = { 80, 50},
6454      Z = { 80, 50},
6455      b = {      , 50},
6456      c = {      , 50},
6457      k = {      , 50},
6458      p = {      , 50},
6459      q = { 50,    },
6460      v = {      , 50},
6461      w = {      , 50},
6462      x = {      , 50},
6463      1 = {100,100},
6464      2 = { 80, 80},
6465      3 = { 80, 50},
6466      4 = { 80, 50},
6467      7 = { 50, 50},
6468      "12 = {500,500},
6469      "13 = {500,500},
6470      ! = {      ,200},
6471      ' = {200,300},
6472      ( = {200,    },
6473      ) = {      ,200},
6474      * = {200,200},
6475      + = {200,250},
6476      - = {200,200},
6477      {,} = {300,300},
6478      . = {400,400},
6479      {=} = {200,200},
6480      : = {      ,200},
6481      ; = {      ,200},
6482      ] = {      ,200}
6483  }
6484
6485  </euf>
6486  </cfg-u>

```

### 15.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym<sup>15</sup>).

```

6487  <*cfg-e>
6488  \SetProtrusion
6489  <zpeu|euroitc> { encoding = U,
6490  <mvs> { encoding = {OT1,U},
6491  <zpeu> family = zpeu }
6492  <euroitc> family = {euroitc,euroitcs} }
6493  <mvs> family = mvs }
6494  {
6495  <zpeu> E = {50, }
6496  <euroitc> E = {100,50}
6497  <mvs> 164 = {50,50}, % \EUR
6498  <mvs> 068 = {50,-100} % \EURdig
6499  }
6500
6501  <*zpeu|euroitc>
6502  \SetProtrusion

```

---

15 Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

Figure 1: Example for interword spacing (from Siemoneit 1989). The numbers indicate the preference/order when the interword space needs to be shrunk.

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

```

6503 { encoding = U,
6504 <zpeu> family = zpeu,
6505 <euroitc> family = {euroitc,euroitcs},
6506 shape = it* }
6507 {
6508 <zpeu> E = {100,-50}
6509 <euroitc> E = {100,}
6510 }
6511
6512 </zpeu|euroitc>
6513 <*zpeu>
6514 \SetProtrusion
6515 { encoding = U,
6516 family = {zpeus,eurosans} }
6517 {
6518 E = {100,50}
6519 }
6520
6521 \SetProtrusion
6522 { encoding = U,
6523 family = {zpeus,eurosans},
6524 shape = it* }
6525 {
6526 E = {200, }
6527 }
6528
6529 </zpeu>
6530 </cfg-e>

```

## 15.9 Interword spacing

Default unit is space.

```

6531 <*m-t>
6532 %%% -----
6533 %%% INTERWORD SPACING
6534
6535 \SetExtraSpacing
6536 [ name = default ]
6537 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6538 {

```

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

6539           { , } = { , -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)

6540           r = { , -300, 300 } ,

- [before or] after lowercase characters with ascenders

6541           b = { , -200, 200 } ,

6542           d = { , -200, 200 } ,

6543           f = { , -200, 200 } ,

6544           h = { , -200, 200 } ,

6545           k = { , -200, 200 } ,

6546           l = { , -200, 200 } ,

6547           t = { , -200, 200 } ,

- [before or] after lowercase characters with x-height plus descender with additional optical space, e. g., ‘v’, or ‘w’

6548           c = { , -100, 100 } ,

6549           p = { , -100, 100 } ,

6550           v = { , -100, 100 } ,

6551           w = { , -100, 100 } ,

6552           z = { , -100, 100 } ,

6553           x = { , -100, 100 } ,

6554           y = { , -100, 100 } ,

- [before or] after lowercase characters with x-height plus descender without additional optical space

6555           i = { , 50, -50 } ,

6556           m = { , 50, -50 } ,

6557           n = { , 50, -50 } ,

6558           u = { , 50, -50 } ,

- after colon and semicolon

6559           : = { , 200, -200 } ,

6560           ; = { , 200, -200 } ,

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

6561           . = { , 250, -250 } ,

6562           ! = { , 250, -250 } ,

6563           ? = { , 250, -250 } ,

The order has to be reversed when enlarging is needed.’

6564        }

6565

Questions are:

- Is the result really better?
- Is it overdone? (Try with a factor < 1000.)

- Should the first parameter also be used? (Probably.)
- What about quotation marks, parentheses etc.?

Furthermore, there seems to be a pdfTeX bug with spacing in combination with a non-zero `\spaceskip` (reported by *Axel Berger*):

```
\parfillskip0pt \rightskip0pt plus 1em \spaceskip\fontdimen2\font
test test\par
\pdfadjustinterwordglue2 \stbscode\font`t=-50
test test
\bye
```

### 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’ (`\fontdimen7`) for Computer Modern Roman is a third of `\fontdimen2`, i.e., 333.

```
6566 \SetExtraSpacing
6567   [ name      = nonfrench-cmr,
6568     load      = default,
6569     context   = nonfrench ]
6570   { encoding = {OT1,T1,LY1,OT4,QX,T5},
6571     family   = cmr }
6572   {
```

latex.ltx has:

```
\def\nonfrenchspacing{
  \sfcode`. 3000
  \sfcode`? 3000
  \sfcode`! 3000
```

```
6573   . = {333,2000,-667},
6574   ? = {333,2000,-667},
6575   ! = {333,2000,-667},
```

```
\sfcode`\: 2000
```

```
6576   : = {333,1000,-500},
```

```
\sfcode`\; 1500
```

```
6577   ; = {    , 500,-333},
```

```
\sfcode`\, 1250
```

```
6578   {,}= {    , 250,-200}
```

```
}
}
```

```
6579 }
6580
```

`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.

```
6581 \SetExtraSpacing
6582 [ name = nonfrench-default,
6583   load = default,
6584   context = nonfrench ]
6585 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6586 {
6587   . = {240,2000,-667},
6588   ? = {240,2000,-667},
6589   ! = {240,2000,-667},
6590   : = {240,1000,-500},
6591   ; = {   , 500,-333},
6592   {,}= {   , 250,-200}
6593 }
6594
```

## 15.10 Additional kerning

Default unit is 1 em.

```
6595 %% -----
6596 %% ADDITIONAL KERNING
6597
```

A dummy list to be loaded when no context is active.

```
6598 \SetExtraKerning
6599 [ name = empty ]
6600 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
6601 { }
6602
```

### 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>16</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.)

```
6603 \SetExtraKerning
6604 [ name = french-default,
6605   context = french,
6606   unit = space ]
6607 { encoding = {OT1,T1,LY1} }
6608 { }
```

---

16 [http://fr.wikipedia.org/wiki/Espace\\_typographique](http://fr.wikipedia.org/wiki/Espace_typographique), 5 July 2007.



```

6609 : = {1000,}, % = \fontdimen2
6610 ; = {500, }, % ~ \thinspace
6611 ! = {500, },
6612 ? = {500, }
6613 }
6614

```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdf<sub>TeX</sub>.

```

6615 \SetExtraKerning
6616 [ name = french-guillemets,
6617   context = french-guillemets,
6618   load = french-default,
6619   unit = space ]
6620 { encoding = {T1,LY1} }
6621 {
6622   \guillemotleft = { ,800}, % = 0.8\fontdimen2
6623   \guillemotright = {800, }
6624 }
6625
6626 \SetExtraKerning
6627 [ name = french-guillemets-OT1,
6628   context = french-guillemets,
6629   load = french-default,
6630   unit = space ]
6631 { encoding = OT1 }
6632 { }
6633

```

### 15.10.2 Turkish

```

6634 \SetExtraKerning
6635 [ name = turkish,
6636   context = turkish ]
6637 { encoding = {OT1,T1,LY1} }
6638 {
6639   : = {167, }, % = \thinspace
6640   ! = {167, },
6641   {=} = {167, }
6642 }
6643
6644 </m-t>
6645 </config>

```

## 16 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.

```

6646 <*test>
6647 \documentclass{article}
6648
6649 %% Here you can specify the font you want to test, using
6650 %% the commands \fontfamily, \fontseries and \fontshape.
6651 %% Make sure to end all lines with a comment character!
6652 \newcommand*\TestFont{%
6653   \fontfamily{ppl}%
6654   \fontseries{b}%
6655   \fontshape{it}% sc, sl
6656 }
6657
6658 \usepackage{ifthen}
6659 \usepackage[T1]{fontenc}

```

```

6660 \usepackage[latin1]{inputenc}
6661 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
6662
6663 \pagestyle{empty}
6664 \setlength{\parindent}{0pt}
6665 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
6666 \newcommand*\testprotrusion[2][\%]
6667   \ifthenelse{\equal{#1}{r}}{\#2}%
6668   lorem ipsum dolor sit amet,
6669   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
6670   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6671   you know the rest%
6672   \ifthenelse{\equal{#1}{l}}{\#2}%
6673   \linebreak
6674   {\fontencoding\encodingdefault}%
6675   \fontseries\seriesdefault}%
6676   \fontshape\shapedefault}%
6677   \selectfont
6678   Here is the beginning of a line, \dotfill and here is its end\linebreak
6679 }
6680 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
6681 \def\stripprefix#1>{}
6682 \newcount\charcount
6683 \begin{document}
6684
6685 \microtypesetup{expansion=false}
6686
6687 {\centering The font in this document is called by:\}
6688 \texttt{\showTestFont}\par\bigskip
6689
6690 \TestFont\selectfont
6691 This line intentionally left empty\linebreak
6692 %% A -- Z
6693 \charcount=65
6694 \loop
6695   \testprotrusion{\char\charcount}
6696   \advance\charcount 1
6697   \ifnum\charcount < 91 \repeat
6698 %% a -- z
6699 \charcount=97
6700 \loop
6701   \testprotrusion{\char\charcount}
6702   \advance\charcount 1
6703   \ifnum\charcount < 123 \repeat
6704 %% 0 -- 9
6705 \charcount=48
6706 \loop
6707   \testprotrusion{\char\charcount}
6708   \advance\charcount 1
6709   \ifnum\charcount < 58 \repeat
6710 %%
6711 \testprotrusion[r]{,}
6712 \testprotrusion[r]{.}
6713 \testprotrusion[r]{;}
6714 \testprotrusion[r]{:}
6715 \testprotrusion[r]{?}
6716 \testprotrusion[r]{!}
6717 \testprotrusion[l]{\textexclamdown}
6718 \testprotrusion[l]{\textquestiondown}
6719 \testprotrusion[r]{}}
6720 \testprotrusion[l]{(}
6721 \testprotrusion{/}
6722 \testprotrusion{\char~\}

```

```
6723 \testprotrusion{-}
6724 \testprotrusion{\textendash}
6725 \testprotrusion{\textemdash}
6726 \testprotrusion{\textquoteleft}
6727 \testprotrusion{\textquoteright}
6728 \testprotrusion{\textquotedblleft}
6729 \testprotrusion{\textquotedblright}
6730 \testprotrusion{\quotesinglbase}
6731 \testprotrusion{\quotedblbase}
6732 \testprotrusion{\guilsinglleft}
6733 \testprotrusion{\guilsinglright}
6734 \testprotrusion{\guillemotleft}
6735 \testprotrusion{\guillemotright}
6736
6737 \newpage
6738 The following displays the current font stretched by 5%,
6739 normal, and shrunk by 5\%:
6740
6741 \bigskip
6742 \newlength{\MTln}
6743 \newcommand*{\teststring}
6744 {ABCDEFGH IJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
6745 \settowidth{\MTln}{\teststring}
6746 \microtypesetup{expansion=true}
6747
6748 \parbox{1.05\MTln}{\teststring\linebreak\\
6749 \teststring}\par\bigskip
6750 \parbox{0.95\MTln}{\teststring}
6751
6752 \end{document}
6753 /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 Change history

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

General: Initial version ..... 1

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

General: configuration file names in lowercase (suggested by <i>Harald Harders</i> ) .....	78	<code>\MT@get@basefamily</code> : only remove suffix if it is 'x' or 'j' .....	79
issue an error instead of a warning, when pdfTeX version is too old for autoexpand .....	121	<code>\MT@get@listname@</code> : don't check for empty attributes list .....	80
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i> ) .....	129	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i> ) .....	42
Protrusion: add factors for some more characters settings for Adobe Minion (contributed by <i>Harald Harders</i> ) .....	135	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i> ) .....	84
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance .....	136	<code>\MT@pdftex@no</code> : fix: version check (reported by <i>Harald Harders</i> ) .....	38
<code>\MT@declare@sets</code> : remove spaces around set name .....	93	<code>\MT@permute</code> : don't use sets for empty encoding ..	107
<code>\MT@DeclareSet</code> : remove spaces around first argument .....	92	<code>\MT@split@codes</code> : fix: allow zero and negative values ..	58
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded ..	78	<code>\MT@use@set</code> : remove spaces around set name .....	97
		<code>\UseMicrotypeSet</code> : remove spaces around first argument .....	97

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

Font sets: declare cmr as an alias of cmr .....	127	changed .....	93
new: allmath and basicmath .....	126	<code>\MT@get@inh@list</code> : fix: set inheritance list \globally to \empty .....	82
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding .....	156	<code>\MT@get@listname@</code> : alternatively check for alias font name .....	80
add settings for Computer Modern Roman math symbols .....	161	<code>\MT@get@size</code> : additional magic to catch some errors hijack \set@fontsize instead of \setfontsize ..	95
<code>\MT@encoding@check</code> : check whether only one encoding specified .....	107	<code>\MT@loop</code> : fix: new macro, used instead of \loop ..	46
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement .....	55	<code>\MT@maybe@do</code> : also check for alias font name .....	55
<code>\MT@get@basefamily</code> : also remove 'w' (swash capitals) .....	79	<code>\MT@permute@@@@@</code> : more sanity checks for \SetProtrusion and \SetExpansion .....	109
<code>\MT@get@highlevel</code> : check whether defaults have		<code>\MT@setupfont</code> : also search for alias font file .....	52
		fix: call \@@enc@update if necessary .....	52

### Version 1.3 (2004/10/27)

General: fix: specifying load option does no longer require to give a name, too .....	103	<code>\MT@fix@catcode</code> : check some category codes (compatibility with german) .....	34
Font sets: declare aer, zer and hfor as aliases of cmr .....	127	<code>\MT@load@list</code> : check whether list exists .....	78

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

General: check for pdfcprot .....	50	(OT1, T1, lmr) .....	140
don't use scratch registers in global definitions ..	82	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options .....	116
use \pickup@font instead of \define@newfont as the hook for \MT@setupfont .....	87	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too .....	100
use one instead of five counters .....	47		
Protrusion: tweak quote characters for cmr variants			

**Version 1.4a (2004/11/17)**

General: new option: final .....	114	when reading files (reported by <i>Michael Hoppe</i> )	79
\MT@cfg@catcodes: fix: reset some more catcodes			

**Version 1.4b (2004/11/26)**

General: fix: set catcodes before reading global configuration file (reported by <i>Christoph Bier</i> ) ..	115	name if encoding failed .....	56
new message if \pdfoutput is changed .....	120	\MT@get@basefamily: fix: failed for font names of the form abczz (reported by <i>Georg Verwey</i> ) .....	79
optimisation: use less \expandafters and \csnames	41	\MT@get@slot: don't define \MT@char globally (save stack problem) .....	82
Protrusion: harmonise dashes in upshape and italic (cmr, pad, ppl) .....	135	\MT@ifdimen: don't set \MT@count globally (save stack problem) .....	43
slanted like italics .....	143	\MT@use@set: don't use undeclared font sets .....	97
\MT@checklist@family: fix: don't try alias family			

**Version 1.5 (2004/12/15)**

General: defaults: step: 4 (suggested by <i>Hàn Thế Thành</i> ) .....	114	\MT@cfg@catcodes: reset catcode of '=' (compatibility with Turkish babel) .....	79
defaults: calculate step as min(stretch,shrink)/5	120	\MT@fix@catcode: reset catcode of '~' (compatibility with chemsym) .....	34
defaults: turn off expansion for DVI output .....	119	\MT@get@highlevel: don't test defaults if called after begin document .....	93
disable automatic expansion for DVI output .....	121	\MT@scale@factor: warning for factors outside limits	60
new option: selected, by default false (suggested by <i>Hàn Thế Thành</i> ) .....	112	\MT@scale@to@em: don't use \lpcode and \rprcode for the calculation .....	59
Documentation: add 'Short history' .....	30	\MT@set@ex@codes: allow non-selected font expansion .....	64
add note about DVIoutput option .....	9	\MT@set@pr@codes: adjust protrusion factors before setting the inheriting characters .....	57
Inheritance: remove \ss from T1 list, add \DJ .....	130		
Protrusion: settings for Bitstream Charter .....	135		
\DeclareMicrotypeAlias: remove spaces around arguments .....	98		

**Version 1.6 (2005/01/24)**

General: defaults: turn off expansion for old pdfTeX versions .....	115	improve settings for numbers (pointed out by <i>Peter Muthesius</i> ) .....	137
disable automatic expansion for old pdfTeX versions .....	121	tune CMR math letters (OML encoding) .....	161
load a font if none is selected .....	52	\MT@get@charwd: use e-TeX's \fontcharwd, if available	59
new option: factor, by default 1000 .....	114	\MT@get@inh@list: correct message if selected is false .....	82
restructure dtx file .....	126	\MT@set@ex@codes: introduce factor option .....	64
test whether \pickup@font has changed .....	90	\MT@set@pr@codes: introduce factor option .....	57
test whether numeric options receive a number .....	114	\MT@use@set: retain current set if new set is undeclared .....	97
use e-TeX's \ifcsname and \ifdefined if defined ..	42	\MT@vinfo: new macro instead of \ifMT@verbose ..	35
Protrusion: add italic uppercase Greek letters ..	143		

**Version 1.6a (2005/02/02)**

Documentation: add table of fonts with tailored protrusion settings .....	21	reported by <i>Bernard Gaulle</i> ) .....	82
\MT@get@slot: completely redone, hopefully more robust (compatible with frenchpro; problem		\MT@pdf@tex@no: new macro .....	37
		\MT@reset@ef@codes: only reset \efcodes for older pdfTeX versions .....	65

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

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i> ) .....	94	fix: remove space after autoexpand .....	105
disallow automatic expansion if pdfTeX too old	105	modify \showhyphens .....	122
		new value for verbose option: errors .....	114

shorter command names	47	test whether <code>\(encoding)\(...</code> is defined	83
warning when running in draft mode	119	<code>\MT@if@list@exists</code> : don't define <code>\MT@#1@c@name</code>	
Documentation: add hint about compatibility	27	globally, here and elsewhere	81
remove table of match order	12	<code>\MT@if@dimen</code> : comparison with 1 to allow size smaller	
Protrusion: fix: remove <code>\</code> from OT1, add		than 1 (suggested by <i>Andreas Böhmann</i> )	43
<code>\textbackslash</code> to T1 encoding	138	<code>\MT@increment</code> : use e-TeX's <code>\numexpr</code> if available	47
<code>\DeclareMicrotypeAlias</code> : may also be used inside		<code>\MT@is@composite</code> : new macro: construct command	
configuration files	98	for composite character; no uncontrolled expansion	85
<code>\LoadMicrotypeFile</code> : new command (suggested by		<code>\MT@scale</code> : new macro: use e-TeX's <code>\numexpr</code> if available	47
<i>Andreas Böhmann</i> )	98	<code>\MT@set@ex@codes</code> : two versions of this macro	64
<code>\Microtype@Hook</code> : new command for font package		<code>\MT@split@name</code> : don't define <code>\MT@encoding &amp;c.</code>	
authors	116	globally	54
<code>\microtypesetup</code> : fix: warning also when setting to		<code>\MT@test@ast</code> : make it simpler	93
(no)compatibility	116	<code>\MT@try@order</code> : always check for size, too (suggested	
<code>\MT@begin@catcodes</code> : also use inside configuration		by <i>Andreas Böhmann</i> )	80
commands	79	fix: also check for <code>//(series)/(shape)//</code> (reported by	
<code>\MT@cfg@catcodes</code> : reset catcode of <code>'</code> (compatibility		<i>Andreas Böhmann</i> )	80
with french* packages)	79	<code>\MT@warn@code@too@large</code> : new macro: type out max-	
<code>\MT@get@listname@</code> : use <code>\etfor</code> ( <i>Andreas Böhmann's</i>		imum protrusion factor	60
idea)	80	<code>\MT@warn@err</code> : new macro: for verbose=errors	35
<code>\MT@get@slot</code> : remove backslash hack	82		
test for <code>\chardef</code> commands	83		

### Version 1.8 (2005/06/23)

General: <code>\SetProtrusion</code> : new key: unit	104	<code>\MT@find@file</code> : no longer wrap names in commands	78
if font substitution has occurred, set up the substi-		<code>\MT@get@charwd</code> : warning for missing (resp. zero-	
tute font, not the selected one	87	width) characters	59
new option: config to load a different main config-		<code>\MT@get@font@dimen@six</code> : new macro: test whether	
uration file	115	<code>\fontdimen 6</code> is defined	57
new option: unit, by default character	114	<code>\MT@get@listname@</code> : made recursive	80
Documentation: add example for factor option	13	<code>\MT@get@slot</code> : fix: expand active characters	82
add example of how to get rid of a widow (sugges-		test whether <code>\(encoding)\(...</code> is defined made more	
ted by <i>Adam Kucharczyk</i> )	15	robust	83
add hint about error messages	27	<code>\MT@get@unit</code> : new macro: get unit for codes	62
Font sets: add U encoding to allmath	126	<code>\MT@in@list</code> : made recursive	46
declare <code>pxr</code> and <code>txr</code> as aliases of <code>ppl</code> resp. <code>ptm</code>	128	<code>\MT@is@active</code> : new macro: translate inputenc-	
Inheritance: remove <code>\DJ</code> from T1 list (it's the same as		defined characters	84
<code>\DH</code> )	130	<code>\MT@is@letter</code> : warning for non-ASCII characters	83
Protrusion: add LY1 characters for Times	142	<code>\MT@led@kern</code> : character protrusion with <code>ledmac</code>	49
settings for AMS math fonts	164	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\@for</code>	45
verified settings for slanted Computer Modern Ro-		<code>\MT@map@tlist@n</code> : new macro: used instead of <code>\@tfor</code>	45
man	149	<code>\MT@old@cmd</code> : renamed commands from	
<code>\add@accent</code> : fix: disable micro-typographic setup in-		<code>\..MicroType..</code> to <code>\..Microtype..</code>	35
side <code>\add@accent</code> (reported by <i>Stephan Hennig</i> )	89	<code>\MT@pdftex@no</code> : case 5: pdfTeX 1.30	37
<code>\DeclareMicrotypeAlias</code> : warning when overriding		<code>\MT@permute@@@@@</code> : add ranges to the beginning of	
an alias font	98	the lists	109
<code>\DeclareMicrotypeSetDefault</code> : new command: set		<code>\MT@scale</code> : fix: remove spaces in e-TeX variant (repor-	
default font set	97	ted by <i>Mark Rossi</i> )	47
<code>\MT@cfg@catcodes</code> : reset catcodes of the remaining		<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when	
ASCII characters	79	hyperref is loaded	50
<code>\MT@check@rlist</code> : made recursive	110	restore csquotes's active characters	50
<code>\MT@curr@list@name</code> : new macro: current list type		restore percent character if Spanish babel is loaded	50
and name	86	<code>\MT@split@codes</code> : get character width once only	58
<code>\MT@declare@sets</code> : warning when redefining a set	93	<code>\MT@use@set</code> : fix: remove braces in first line	97
<code>\MT@define@set@key@</code> : use comma lists instead of		<code>\MT@xadd</code> : simplified	44
token lists	93		

### Version 1.9 (2005/10/28)

General: <code>\DeclareMicrotypeSet</code> : new key: font	95	<code>\SetProtrusion</code> : value 'relative' renamed to	
--	----	---	--

‘character’ for key unit	104	settings for T5 encoded Computer Modern Roman	135
allow context-specific font setup	87	\DisableLigatures: new command: disable ligatures	
disable expansion if both step and shrink are zero	121	(requires pdf $\TeX$ 1.30)	99
disable microtype setup inside hyperref’s		\microtypecontext: new command: change setup	
\pdfstringdef (reported by <i>Hàn Thế Thành</i> )	51	context in the document	91
fix: use true as the default value	111	\MT@checklist@family: fix: add two missing	
option unit: rename value relative to character	114	\expandafters	56
warning if user requested zero step	120	\MT@detokenize@c: fix the $\mathbb{X}$ version	42
Documentation: add hint about verbatim environment	26	\MT@exp@two@n: new macros: less \expandafters	42
add remark about Type 1 fonts required for automatic font expansion	8	\MT@get@opt: new key ‘preset’ to set all characters to the specified value before loading the lists	62
Font sets: add OT4 encoding to text sets	126	\MT@is@active: redone: use \set@display@protect	84
add T5 encoding to text sets	126	\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning	83
declare qpl and qtm (qfonts, $\TeX$ Gyre) as aliases of ppl resp. ptm	128	\MT@maybe@do: redone	55
Inheritance: add list for OT4	131	\MT@pdfTeX@no: compatibility with $\TeX$ Live hack (reported by <i>Herbert Voß</i> )	38
add list for T5 (requested by <i>Hàn Thế Thành</i> )	132	\MT@rem@from@clist: new macro: remove an item from a comma list	46
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR	139	\MT@scale@factor: generalised	60
settings for OT4 encoding (Computer Modern Roman, Palatino, Times)	135	\MT@toks: use instead of \toks@	39
		\SetProtrusion: (et al.) new key: font	99

### Version 1.9a (2005/12/05)

General: ‘(file name)/(line number)’ as default list name	103	diately (requested by <i>Georg Verwey</i> )	93
new option: defer setup, by default true	113	\MT@get@highlevel: no longer check whether defaults have changed	93
remove superfluous test whether \pickup@font has changed	90	\MT@ifdefined@c@T: new macros: true case only	42
Documentation: add explanation for error message in DVI mode	27	\MT@ifint: use \pdfmatch if available	43
add explanation for error message with non-Type 1 fonts	27	\MT@ifstreq: use \pdfstrcmp if available	44
Font sets: declare mdbch (mathdesign) as an alias of Charter	128	\MT@in@clist: fix	45
Protrusion: fix: remove ‘_’ from OT1 encoding	139	\MT@info@missing@char: info instead of warning (after <i>Michael Hoppe</i> reported that the ‘fl’ ligature is missing in Palatino SC)	60
settings for T5 encoded Charter	135	\MT@is@feature: new macro: check for pdf $\TeX$ feature	48
\microtypesetup: inside the preamble, accepts all package options	116	\MT@map@clist@n: following $\LaTeX$ 3	45
\MT@check@font@cx: optimise context-sensitive setup	90	\MT@permute@@@@: don’t define permutations for unused encodings	108
\MT@define@set@key@: don’t expand variables immediately		\MT@rem@from@clist: fix	46
		\MT@setup@: defer setup until the end of the preamble	48

### Version 1.9b (2006/01/20)

General: compatibility with listings: sanitise more catcodes (reported by <i>Holger Uhr</i> )	51	add samples of micro-typographic features	4
compatibility with the extendedchar option of the listings package	51	\MT@features: use throughout the package to adjust to beta-ness	48
Documentation: activate expansion in the distributed PDF	1	\MT@ifdimen: use \pdfmatch if available	43
		\MT@warn@code@too@large: fix calculation with present factor	60

### Version 1.9c (2006/02/02)

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verwey</i> )	22	\MT@define@code@key@font: fix: context was ignored	102
Protrusion: settings for URW Garamond	136	\MT@define@code@key@size: fix: embrace \MT@tempsize in \csname (bug introduced in v1.9b)	102



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

Font sets: md* instead of m series in basic sets . . . .	126	\MT@get@font@dimen: warning for zero fontdimen . .	60
add QX encoding to text sets . . . . .	126	\MT@get@opt: optimise: don't reset when preset op-	
Inheritance: add list for QX encoding (contributed by		tion is set . . . . .	62
Maciej Eder) . . . . .	131	set list name before presetting . . . . .	62
Protrusion: settings for QX encoding (contributed by		\MT@is@active: support for Unicode (inputenc/utf8) .	84
Maciej Eder) . . . . .	141	\MT@setupfont@hook: restore % and # when tex4ht	
settings for Euro symbols (Adobe, ITC, marvosym)	172	is loaded (reported by <i>Peter Dybala</i> ) . . . . .	50
tweak AMS settings . . . . .	164	\SetProtrusion: (et al.) optimise: unify keys for	
\DeclareCharacterInheritance: fix: empty context	106	mandatory argument . . . . .	99
\MT@detokenize@n: new macro: use \detokenize if		(et al.) split keys of optional and mandatory argu-	
available . . . . .	42	ment . . . . .	99
\MT@get@ex@opt: fix: evaluate preset . . . . .	65		

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

General: fix: default value for activate: true . . .	112	settings for Euler Roman font . . . . .	168
Documentation: add hint about unknown encodings	26	\DeclareCharacterInheritance: new key 'inputenc'	
include LPPL . . . . .	194	to set the input encoding . . . . .	106
Font sets: declare zeur and zeus (eulervm) as aliases		\MT@rem@from@cllist: model after \@removeelement	46
of eur resp. eus (euler) . . . . .	128	\MT@setup@: empty \MT@setup@ after use (compatibil-	
Inheritance: adapt to marvosym's changed encoding	133	ity with the combine class) . . . . .	48
Protrusion: complete settings for Euler Fraktur and		\pickup@font: no tracing with trace package . . . .	89
Script fonts . . . . .	171	\SetExpansion: new key: inputenc . . . . .	100
fix: forgotten comma in mt-mvs.cfg; adapt to		\SetProtrusion: (et al.) new key: inputenc . . . .	99
marvosym's changed encoding . . . . .	172		

**Version 1.9f (2006/09/09)**

Protrusion: fix: euler-vm did not load euler settings	169	\MT@reset@context: only reset context if it has actu-	
\MT@curr@list@name: fix: \MessageBreak must not be		ally been changed . . . . .	91
expanded . . . . .	86	\MT@set@inh@list: fix: forgotten comma in the fea-	
\MT@gdef@n: new macros: global variants . . . . .	41	tures list . . . . .	106
\MT@get@inh@list: fix: input encoding must be set		\MT@set@named@keys: new macro: set name first, sim-	
after the inheritance list has been parsed . . . .	82	plify parsing of optional argument . . . . .	101
\MT@glet: new macro . . . . .	41	\SetProtrusion: (et al.) set catcodes before parsing	
		optional argument . . . . .	99

**Version 2.0 (2007/01/14)**

General: (beta:1) new option: babel, by default false		Documentation: add hint about how to increase	
(language-dependent setup suggested by <i>Ulrich</i>		font_max and font_mem_size . . . . .	28
<i>Dirr</i> ) . . . . .	112	add hint about warning when tracking and expan-	
(beta:1) new option: letterspace, by default 100	114	sion is applied to a font . . . . .	28
(beta:3) compatibility with soul: register \textls		add remark about 'draft' option disabling	
and \lsstyle . . . . .	51	microtype (noted by <i>Michalis Miatidis</i> ) . . . . .	9
(beta:8) option 'babel': fix: switch off French		qualify hint about web documents with regard to	
babel's shorthands properly (reported by <i>Daniel</i>		older pdfTeX versions . . . . .	25
<i>Flipo</i> ) . . . . .	125	qualify hints about expansion error messages with	
(beta:8) option 'babel': switch off Turkish babel's		regard to older pdfTeX versions . . . . .	27
shorthands . . . . .	125	Font sets: add footnotesize and scriptsize sets	127
compatibility with listings: set catcode of back-		add smallcaps set . . . . .	127
slash to zero (reported by <i>Steven Bath</i> ) . . . . .	51	\DeclareMicrotypeBabelHook: (beta:1) new com-	
maybe disable \MT@no ligatures after the preamble	124	mand: interaction with babel . . . . .	99
new package letterspace: a stripped-down ver-		\lsstyle: (beta:1) new command: letterspacing . .	72
sion, containing the letterspacing commands		(beta:8) fix: font switches don't pose a problem	
only . . . . .	1	anymore . . . . .	72
option 'unit', \SetProtrusion: deprecate value		(beta:8) fix: letterspacing commands may be nes-	
'relative' completely . . . . .	104	ted . . . . .	72



totally redone, using the new <code>\letterspacefont</code>	72	<code>\SetExtraKerning</code> : (beta:1) new command: additional kerning	101
<code>\MT@declare@sets</code> : fix: empty size list when redefining set	93	<code>\SetExtraSpacing</code> : (beta:1) new command: adjustment of interword spacing	101
<code>\MT@is@symbol</code> : made even more robust	85	<code>\SetTracking</code> : new command: tracking	100
<code>\MT@load@inputenc</code> : sanitise catcodes before loading input encoding (problem with listings)	63	<code>\textls</code> : (beta:1) new command: letterspacing	76
<code>\MT@pdfTeX@no</code> : (beta:1) case 6: pdfTeX 1.40	37	starred version: remove spaces around text	76
<code>\MT@split@name</code> : adjust to possible letterspacing	54	<code>\tracingmicrotypeinpdf</code> : new debug method: mark all fonts with PDF annotations	36

## Version 2.1 (2007/01/21)

General: compatibility with CJK: also check for its definition	88	<code>\lslig</code> : new command: protect ligatures in letterspaced text	72
compatibility with pinyin: disable microtype in <code>\py@macron</code> (reported by <i>Sven Naumann</i> )	52	<code>\MT@get@ls@basefont</code> : redone: use <code>\pdfmatch</code> to make it bullet-proof	73
fix: letterspace package forgot to load <code>keyval</code>	39	<code>\textls</code> : fix: use <code>\hmode@bgroup</code>	76

## Version 2.2 (2007/07/14)

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i> )	48	fix: $\epsilon$ -TeX version shouldn't use <code>\x</code> and <code>\y</code> (found by <i>Wiebke Petersen</i> )	44
new option: <code>copyfonts</code>	113	<code>\MT@is@composite</code> : more robust: expand exactly once	85
simplify key declarations	103	<code>\MT@is@symbol</code> : expand once more (for frenchpro)	85
warning if stretch or shrink aren't multiples of step	122	<code>\MT@lsfont</code> : use <code>\font@name</code> , not <code>\MT@font</code>	70
Documentation: add hint about error message with pdfTeX 1.40	27	<code>\MT@maybe@etex</code> : use catcode trickery	40
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i> )	26	<code>\MT@pdfTeX@no</code> : case 7: pdfTeX 1.40.4	37
add overview	6	<code>\MT@preset@aux@space</code> : generalised	63
logo transparency and amusement	1	<code>\MT@requires@luatex</code> : (basic) support for luaTeX	40
Font sets: declare FPL Neu as an alias of Palatino	128	<code>\MT@set@all@pr</code> : (et al.) allow empty values	58
declare <code>chr</code> ( <code>chmath</code> ) as an alias of Charter (reported by <i>Geoff Vallis</i> )	128	<code>\MT@set@inputenc</code> : only load <code>inputenc</code> files if necessary	62
default set for tracking: <code>smallcaps</code>	127	<code>\MT@set@tr@codes</code> : disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	70
Inheritance: remove <code>'-'</code> $\rightarrow$ <code>'127'</code>	130	possibility to customise interword spacing	70
Protrusion: settings for Bitstream Letter Gothic	135	<code>\MT@setupfont</code> : don't call <code>\@enc@update</code> anymore	53
Spacing: add sample	173	only add features that are available with the respective pdfTeX	53
Tracking: add ligatures that are to be disabled	133	<code>\MT@setupfont@hook</code> : restore percent character if Galician babel is loaded	50
<code>\DeclareMicrotypeVariants</code> : new command	98	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letterspaced fonts	58
<code>\DisableLigatures</code> : new optional argument: disable selected ligatures only	99	<code>\MT@tracking</code> : remember fonts that shouldn't be letterspaced	69
<code>\lslig</code> : always defined	72	<code>\MT@tracking@</code> : fix: tracking couldn't be re-enabled	69
<code>\MT@checklist@font</code> : fix: construct font name from characteristics	57	<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	123
<code>\MT@copy@font</code> : optionally work on copies of fonts	53	<code>\MT@with@babel@and@T</code> : also inspect class options	49
<code>\MT@get@basefamily</code> : redone, working on font names and suffixes of arbitrary length	79	<code>\pickup@font</code> : letterspace: setup inside group	89
<code>\MT@get@charwd</code> : subtract letterspacing amount from width	59	<code>\SetTracking</code> : new key 'no ligatures' to disable ligatures of letterspaced fonts	100
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	73	new keys 'spacing' and 'outer spacing' to adjust interword spacing (suggested by <i>Steven E. Harris</i> )	100
<code>\MT@ifdimen</code> : employ luaTeX features if available	43	third argument may be empty	100
<code>\MT@ifint</code> : employ luaTeX features if available	43	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code>	91
<code>\MT@ifstreq</code> : employ luaTeX features if available	44		

**Version 2.3 (2007/12/23)**

General: disable <code>\microtypecontext</code> in <code>hyperref</code> 's <code>\pdfstringdef</code> .....	51	<code>\MT@in@cllist</code> : fix: don't use <code>\x</code> (reported by <i>Peter Meier</i> ) .....	45
enable protrusion when tracking is enabled ...	123	<code>\MT@is@active</code> : support for extended Unicode ( <code>inputenc/utf8x</code> resp. <code>ucs</code> ) – experimental ...	84
fix: really switch off Turkish shorthands .....	125	<code>\MT@noligatures</code> : fix: set evaluation didn't work (bug introduced in 2.2) .....	77
new value for verbose option: silent (suggested by <i>Karl Berry</i> ) .....	114	<code>\MT@plain</code> : letterspace: support for <code>eplain/miniltx</code> ..	37
turned some warnings into errors .....	114	<code>\MT@set@curr@os</code> : adjusting spaces made more reli- able .....	71
Documentation: add kerning sample .....	20	<code>\MT@set@tr@codes</code> : also adjust tracking if protrusion is not enabled, and even for letterspace (reported by <i>Stephan Hennig</i> ) .....	70
add letterspacing illustration .....	17	possibility to customise outer kerning (suggested by <i>Stephan Hennig</i> ) .....	70
<code>\do@subst@correction</code> : remember substitute font for all times (reported by <i>Stephan Hennig</i> ) .....	89	<code>\MT@SetTracking</code> : sanity check for value .....	100
<code>\slig</code> : redone: extract outer kerns from current letterspacing amount .....	72	<code>\MT@tr@outer@l</code> : only change pre outer space if it contains shrink .....	74
<code>\microtypecontext</code> : made robust (reported by <i>Ste- phan Hennig</i> ) .....	91		
<code>\MT@begin@catcodes</code> : fix: don't disable <code>\KV@sp@def</code> ..	79		
<code>\MT@define@set@key@font</code> : font: single asterisk means normal font .....	95		

**Version 2.3a (2008/02/29)**

General: error messages if <code>pdfTeX</code> is too old for exten- sions .....	124	<code>\MT@fix@catcode</code> : fix catcodes earlier, and also for the letterspace package .....	34
fix test for <code>soul</code> under plain <code>TeX</code> .....	51	<code>\MT@getkey</code> : fix: <code>key=val</code> in class options list ....	119
Documentation: add hint about <code>babel</code> having to be loaded first .....	27	<code>\MT@maybe@etex</code> : removed .....	40
add table of available and enabled features .....	7	<code>\MT@set@codes</code> : generalised .....	58
mention <code>soulutf8</code> .....	29	<code>\MT@setupfont@hook</code> : restore percent character if Mexican <code>babel</code> is loaded .....	50
Protrusion: adjust LMR quotation marks again ....	141		

**Version 2.3b (2008/06/04)**

General: compatibility with <code>CJKutf8</code> : also check for its definition .....	88	known slots .....	77
<code>\MT@exp@gcs</code> : new macro: reduce save stack size ..	41	<code>\MT@requires@latex</code> : new macro .....	37
<code>\MT@font@copy</code> : enable font copies also with protru- sion contexts (reported by <i>Nathan Rosenblum</i> ) ..	53	<code>\MT@set@tr@codes</code> : fix: protrusion adjustment only for new fonts (reported by <i>Wolfram Schaalo</i> ) ..	70
<code>\MT@get@size</code> : grouping .....	95	<code>\MT@tr@outer@l</code> : fix: only in horizontal mode ....	74
<code>\MT@noligatures@</code> : fix: warning messages for un-		make <code>\spaceskip-aware</code> ( <code>ragged2e</code> ) .....	74
		<code>\MT@tr@outer@r@</code> : additional test for horizontal mode	75

**Version 2.3c (2008/11/11)**

General: <code>luaTeX</code> supported by default .....	41	coding (reported by <i>Vasile Gaburici</i> ) .....	132
Documentation: add hint about partial incompatibil- ity with <code>CJK</code> .....	27	<code>\MT@detokenize@</code> : fix: remove last space only (repor- ted by <i>Ulrich Dirr</i> ) .....	42
add hint about spacing being experimental ....	26	<code>\MT@tr@outer@r@</code> : additional test for horizontal mode (reported by <i>Sveinung Heggen</i> ) .....	75
Inheritance: add <code>\textcommabelow[STst]</code> to QX en-			

## B Index

Links (in blue) refer to the page where the corresponding entry is described (bold face) resp. occurs. Plain numbers (in black) refer to the code line where the corresponding entry is defined (underlined) resp. used.

### Options:

DVIoutput	9
activate	6
auto	8
babel	9
config	9
draft	9
expansion	6
factor	7
final	9
kertering	7
letterspace	9
protrusion	6
selected	8
shrink	8
spacing	7
step	8
stretch	8
tracking	7
unit	8
verbose	9

### User Commands:

<code>\DeclareCharacterInheritance</code>	20
<code>\DeclareMicrotypeAlias</code>	22
<code>\DeclareMicrotypeBabelHook</code>	23
<code>\DeclareMicrotypeSet</code>	10
<code>\DeclareMicrotypeSet*</code>	10
<code>\DeclareMicrotypeSetDefault</code>	11
<code>\DeclareMicrotypeVariants</code>	21
<code>\DeclareMicrotypeVariants*</code>	21
<code>\DisableLigatures</code>	25
<code>\LoadMicrotypeFile</code>	22
<code>\lslig</code>	24
<code>\lstyle</code>	24
<code>\microtypecontext</code>	22
<code>\microtypesetup</code>	10
<code>\SetExpansion</code>	14
<code>\SetExtraKerning</code>	19
<code>\SetExtraSpacing</code>	18
<code>\SetProtrusion</code>	13
<code>\SetTracking</code>	16
<code>\textls</code>	24
<code>\textls*</code>	24
<code>\textmicrotypecontext</code>	22
<code>\UseMicrotypeSet</code>	11

### A

a0poster (package)	95
activate (option)	6, 112, 184
<code>\add@accent</code>	2325
ae (package)	21, 127
amssymb (package)	164
article (package)	23
auto (option)	8, 112

### B

babel (option)	9, 23, 27, 30, 112, 184
----------------	-------------------------

babel (package)	3, 5, 17, 23, 24, 27, 50, 79, 99, 125, 126, 128, 181, 182, 184–186
-----------------	---

### C

chemsym (package)	181
chmath (package)	21, 128, 185
CJK (package)	27, 52, 88, 185, 186
CJKutf8 (package)	88, 186
cm-super (package)	8
color (package)	9, 120
combine (package)	48, 184
config (option)	9, 21, 31, 115, 182
contour (package)	120
copyfonts (option)	53, 112, 113, 185
crop (package)	120
csquotes (package)	29, 50, 139, 182
<code>\curr@fontshape</code>	2272, 2274, 2276, 2283, 2323

### D

<code>\DeclareCharacterInheritance</code>	20, 41, 3044
<code>\DeclareMicrotypeAlias</code>	22, 43, 70, 2701
<code>\DeclareMicrotypeBabelHook</code>	23, 45, 64, 2754
<code>\DeclareMicrotypeSet</code>	10, 32, 57, 59, 71, 2432
<code>\DeclareMicrotypeSet*</code>	10, 2432
<code>\DeclareMicrotypeSetDefault</code>	11, 34, 61, 2660
<code>\DeclareMicrotypeVariants</code>	21, 42, 58, 63, 2688
<code>\DeclareMicrotypeVariants*</code>	21
deferssetup (option)	50, 112, 113, 116, 183
<code>\define@newfont</code>	2265, 2273, 2277, 2285
<code>\DisableLigatures</code>	25, 40, 62, 2734
<code>\do@subst@correction</code>	2322
docstrip (package)	33
draft (option)	9, 112, 114, 184
dsfont (package)	57
DVIoutput (option)	9, 112, 113, 181

### E

eco (package)	21, 127
<code>\efcode</code>	1285, 1309, 1310, 1350, 1352
eplain (package)	25, 30, 33, 37, 48, 119, 186
eucal (package)	169
eufrak (package)	171
euler (package)	86, 168, 184
eulervm (package)	21, 128, 169, 184
euroitc (package)	172
europs (package)	172
eurosans (package)	172
expansion (option)	6, 7, 11, 111, 112, 119

### F

<code>\f@family</code>	2279, 2280
<code>\f@size</code>	2272, 2274, 2276, 2283, 2323
factor (option)	7, 13, 32, 114, 181
fancyvrb (package)	27
final (option)	9, 32, 112, 114, 181
fix-cm (package)	54
<code>\font</code>	787, 1643

- `\font@name` ..... 109, 823, 825, 827, 829,  
     839, 840, 1533, 1538, 1572, 1579, 1589,  
     1592, 1631, 1643, 1655, 1657, 1662, 1664,  
     1667, 1696, 1751, 1844, 1846, 2265, 2273,  
     2282, 2285, 2306, 2308, 2310, 2324, 2336, 2356  
`\fontcharwd` ..... 1049  
`\fontdimen` ..... 978,  
     985, 1070, 1075, 1174, 1177, 1550, 1551,  
     1579, 1655, 1689, 1690, 1697, 1698, 1704,  
     1708, 1715, 1725, 1729, 1732, 1751, 1857, 1860  
`fontinst` (package) ..... 176  
`fontinstallationguide` (package) ..... 154  
`french` (package) ..... 182  
`frenchpro` (package) ..... 85, 181, 185
- G**
- `german` (package) ..... 34, 180  
`graphics` (package) ..... 9, 120
- H**
- `hfoldsty` (package) ..... 21, 127  
`hyperref` (package) . 9, 50, 51, 87, 120, 182, 183, 186
- I**
- `IEEEtran` (package) ..... 87  
`\iffontchar` ..... 1080  
`\ifMT@auto` ..... 219, 3705, 3752  
`\ifMT@babel` ..... 219, 3877, 3924  
`\ifMT@do` ..... 859, 923, 1512, 1876  
`\ifMT@document` ..... 261, 3160  
`\ifMT@draft` ..... 219, 3643  
`\ifMT@expansion` ..... 219, 3673, 3749  
`\ifMT@if@` ..... 218,  
     690, 696, 714, 745, 2350, 3263, 3944, 3947  
`\ifMT@inannot` ..... 98  
`\ifMT@inlist@` ..... 519, 564, 623,  
     647, 802, 890, 907, 916, 934, 952, 1510,  
     1927, 1938, 1987, 2035, 2344, 2406, 2418, 2719  
`\ifMT@kerning` ..... 219, 3817, 3866, 3937  
`\ifMT@noligatures` ..... 219, 3893  
`\ifMT@nonselected` ..... 1263, 2048, 2056  
`\ifMT@noreset` ..... 2110, 2117  
`\ifMT@opt@auto` ..... 3270, 3708  
`\ifMT@opt@DVI` ..... 3270, 3659  
`\ifMT@opt@expansion` ..... 3270, 3654  
`\ifMT@protrusion` ..... 219, 653, 675, 3661, 3800  
`\ifMT@selected` ..... 219, 3735, 3755  
`\ifMT@spacing` ..... 219, 3806, 3860, 3876  
`\ifMT@tracking` ..... 219, 3795, 3854  
`ifpdf` (package) ..... 9, 88, 120  
`\iftracingmicrotypeinpdfall` ..... 103  
`inputenc` (package) ..... 14,  
     27, 31, 51, 62, 63, 82, 84, 86, 182, 184–186
- J**
- `jurabib` (package) ..... 87
- K**
- `kerning` (option) ..... 7, 11, 30, 112  
`keyval` (package) ..... 39, 58, 129, 185  
`\knaccode` ..... 1465, 1466, 1476, 1479, 1485  
`\knbccode` ..... 1458, 1459, 1475, 1478, 1484  
`\knbscode` ..... 1379, 1380, 1403, 1407, 1413
- L**
- `ledmac` (package) ..... 29, 31, 49, 87, 182  
`ledpar` (package) ..... 29  
`\leftmarginkern` ..... 661  
`letterspace` (option) . 9, 18, 24, 30, 43, 69, 114, 184  
`letterspace` (package) .....  
     ... 1, 24, 25, 30, 33, 37, 49, 70, 72, 184–186  
`\letterspacefont` ..... 1538  
`lineno` (package) ..... 49  
`listings` (package) ..... 27, 51, 63, 79, 183–185  
`lmodern` (package) ..... 8, 141  
`\LoadMicrotypeFile` ..... 22, 44, 73, 2715, 3648  
`\lcode` ..... 991, 1023, 1024, 1210, 1213, 1563  
`\slig` ..... 24, 54, 1648, 1651  
`\lsstyle` ..... 24, 51,  
     737, 760, 765, 1632, 1807, 1810, 1824, 3651
- M**
- `marvosym` (package) ..... 21, 31, 33, 133, 172, 184  
`mathdesign` (package) ..... 21, 128, 183  
`memoir` (package) ..... 23, 87  
`\Microtype@Hook` ..... 116, 3500  
`\microtypecontext` .....  
     ... 22, 47, 735, 2376, 2386, 3650, 3907, 3910  
`microtypecontext` (environment) ..... 22  
`\microtypesetup` ..... 10, 46, 3504, 3649  
`miniltx` (package) ..... 25, 30, 33, 37, 48, 186  
`minimal` (package) ..... 52  
`\MT@font` ..... 316, 805,  
     821, 980, 1071, 1084, 1523, 1990, 2061, 2311  
`\MT@abbr@ex` ..... 599  
`\MT@abbr@ex@c` ..... 599  
`\MT@abbr@ex@inh` ..... 599  
`\MT@abbr@kn` ..... 599  
`\MT@abbr@kn@c` ..... 599  
`\MT@abbr@kn@inh` ..... 599  
`\MT@abbr@nl` ..... 599  
`\MT@abbr@pr` ..... 599  
`\MT@abbr@pr@c` ..... 599  
`\MT@abbr@pr@inh` ..... 599  
`\MT@abbr@sp` ..... 599  
`\MT@abbr@sp@c` ..... 599  
`\MT@abbr@sp@inh` ..... 599  
`\MT@abbr@tr` ..... 599  
`\MT@abbr@tr@c` ..... 599  
`\MT@active@features` .....  
     831, 2338, 2341, 2353, 2360, 2395, 2405,  
     2741, 3662, 3750, 3797, 3808, 3819, 3899, 3900  
`\MT@addto@annot` ..... 98  
`\MT@addto@setup` ..... 638, 640, 699,  
     1187, 2259, 2376, 2377, 2939, 2957, 3374,  
     3505, 3642, 3793, 3853, 3892, 3898, 3923, 3956  
`\MT@auto` ..... 1272, 3704, 3718, 3728, 3732  
`\MT@auto@` ..... 1272, 1281, 1323  
`\MT@autofalse` ..... 221, 3716, 3727  
`\MT@autotruer` ..... 221, 3464  
`\MT@babelfalse` ..... 228  
`\MT@babeltrue` ..... 228  
`\MT@begin@catcodes` 1928, 1929, 1969, 2723, 2736,  
     2761, 2775, 2797, 2816, 2830, 3049, 3477, 3478  
`\MT@cat` ..... 1183, 2251, 2253  
`\MT@cfig@catcodes` ..... 747, 1205, 1954, 1971

- \MT@char 1013, 1023, 1024, 1029, 1030, 1033, 1035,  
 1049, 1050, 1080, 1210–1214, 1309, 1310,  
 1312, 1313, 1350–1352, 1379, 1380, 1386,  
 1387, 1393, 1394, 1397, 1398, 1403–1408,  
 1458, 1459, 1465, 1466, 1469, 1470, 1475–  
 1479, 1887, 2090, 2106, 2107, 2112, 2183,  
 2186, 2188, 2214, 2217, 3109, 3110, 3115, 3116  
 \MT@char@ ..... 2090, 2096, 2101, 2106, 2120, 2122, 2128,  
 2129, 2131, 2144, 2145, 2148, 2149, 2152,  
 2153, 2157, 2159, 2187, 2200, 2203, 2206, 2225  
 \MT@charstring ..... 2103, 2186, 2191  
 \MT@check@active@set ..... 3492, 3668, 3770, 3799, 3811, 3823  
 \MT@check@font ..... 801, 2335, 2397  
 \MT@check@font@cx ..... 2339, 2397  
 \MT@check@range ..... 3234, 3236  
 \MT@check@range@ ..... 3236, 3237  
 \MT@check@rlist ..... 3187, 3227  
 \MT@check@rlist@ ..... 3227, 3228  
 \MT@check@step ..... 3758  
 \MT@checklist@ ..... 867, 883, 1873  
 \MT@checklist@family ..... 901  
 \MT@checklist@font ..... 945  
 \MT@checklist@size ..... 929  
 \MT@checksetup ..... 3514, 3525, 3532, 3551, 3571  
 \MT@clear@options ..... 190, 213, 298, 634, 3636  
 \MT@clist@break ..... 491, 896, 924, 940, 954  
 \MT@clist@function ..... 491  
 \MT@cnt@encoding ..... 3126, 3134, 3135  
 \MT@cnt@family ..... 3132, 3141, 3142  
 \MT@cnt@series ..... 3139, 3148, 3149  
 \MT@cnt@shape ..... 3146, 3154, 3155  
 \MT@config@file 3467, 3475, 3476, 3480, 3481, 3484  
 \MT@context ..... 2011, 2023, 2042, 2069  
 \MT@copy@font ..... 790,  
 818, 2935, 2941, 2953, 2960, 3391, 3393, 3756  
 \MT@copy@font@ .. 818, 2935, 2941, 2953, 2960, 3391  
 \MT@count ..... 429, 430, 583, 1042, 1049, 1051,  
 1052, 1057, 1062, 1063, 1067, 1075, 1106, 1168  
 \MT@curr@file ..... 1932, 1933,  
 1942, 1943, 2725, 2726, 2908, 3071, 3480, 3958  
 \MT@curr@list@name ... 1073, 1109, 1199, 1319,  
 1884, 1913, 2080, 2220, 2228, 2233, 2241, 2247  
 \MT@curr@ls ..... 1569, 1652, 1655  
 \MT@curr@set@name .....  
 2457, 2459–2462, 2464, 2465, 2470, 2475,  
 2479, 2480, 2505, 2509, 2538, 2568, 2574,  
 2580, 2584, 2585, 2812, 2870, 2888, 2903,  
 2915, 2979, 2983, 2988, 2991, 2993, 2997,  
 3000, 3003, 3013, 3026, 3032, 3040, 3072, 3074  
 \MT@declare@char@inh ..... 3054, 3059, 3069  
 \MT@declare@sets ..... 2440, 2446, 2458, 2743  
 \MT@DeclareSet ..... 2435, 2437, 2454  
 \MT@DeclareSetAndUseIt ..... 2434, 2453  
 \MT@DeclareVariants ..... 2691, 2692, 2694  
 \MT@def@bool@opt ..... 3345,  
 3358–3360, 3369, 3381, 3389, 3398, 3405  
 \MT@def@n ..... 305, 2428, 2429  
 \MT@default@ex@set ..... 2676  
 \MT@default@kn@set ..... 2676  
 \MT@default@pr@set ..... 2676  
 \MT@default@sp@set ..... 2676  
 \MT@default@tr@set ..... 2676  
 \MT@define@code@key . 2852, 2895–2898, 3094–3096  
 \MT@define@code@key@font ..... 2875, 2900, 3098  
 \MT@define@code@key@size ..... 2863, 2899, 3097  
 \MT@define@opt@key .. 2901, 2917–2920, 3036–3038  
 \MT@define@optionX ..... 3504, 3560, 3561  
 \MT@define@optionX@ ..... 3562, 3590, 3592, 3593  
 \MT@define@set@key@ ..... 2468, 2624–2627  
 \MT@define@set@key@font ..... 2572, 2629  
 \MT@define@set@key@size ..... 2498, 2628  
 \MT@detokenize@c ..... 346, 2098, 2184  
 \MT@detokenize@n ..... 346, 2216  
 \MT@dimen@six 977, 1043, 1067, 1106, 1724, 1727, 1853  
 \MT@dinfo ..... 86  
 \MT@dinfo@list ..... 880, 891, 894, 899, 908,  
 911, 917, 919, 927, 935, 938, 943, 953, 956, 960  
 \MT@dinfo@nl ..... 86  
 \MT@DisableLigatures ..... 2734  
 \MT@do@font ..... 581, 993, 1285, 1416, 1486, 1563  
 \MT@documentfalse ..... 261  
 \MT@documenttrue ..... 261, 3902  
 \MT@dofalse ..... 859, 871, 895, 912, 939, 957  
 \MT@dotrue ..... 859, 862, 892, 909, 918, 936, 1868  
 \MT@draftfalse ..... 224, 3407  
 \MT@drafttrue ..... 224, 3409  
 \MT@edef@n ..... 307, 2047, 2073,  
 2422, 2858, 2908, 2910, 3070, 3085, 3214, 3438  
 \MT@encoding ..... 849, 949,  
 2006, 2018, 2098, 2184, 2215, 2232, 2239, 2247  
 \MT@encoding@check ..... 3083, 3087  
 \MT@end@catcodes ..... 1930, 1973, 2731,  
 2745, 2772, 2794, 2813, 2827, 2841, 3064, 3479  
 \MT@error ..... 75, 288, 626, 702,  
 1808, 1905, 1918, 2536, 2655, 2682, 2749,  
 2927, 2944, 2969, 3383, 3400, 3537, 3555,  
 3579, 3709, 3721, 3787, 3855, 3861, 3867, 3919  
 \MT@ex@c@name ..... 1256,  
 1258, 1325, 1326, 1336, 1342, 1343, 1355, 2778  
 \MT@ex@c@context ..... 821, 2389, 2427  
 \MT@ex@doc@contexts ..... 2427  
 \MT@ex@factor ..... 229, 1273, 1329  
 \MT@ex@factor@ . 1273, 1287, 1290, 1299, 1300, 1323  
 \MT@ex@inh@name ..... 1311–1313  
 \MT@ex@level ..... 229, 3751, 3753  
 \MT@ex@max ..... 241, 1302, 1303  
 \MT@ex@min ..... 241, 1305, 1306  
 \MT@ex@set@name ..... 2647  
 \MT@ex@split@val ..... 1297  
 \MT@exp@cs ..... 302, 305, 308,  
 311, 314, 832, 834, 843, 933, 1034, 1093,  
 1096, 1126, 1130, 1140, 1143, 1146, 1313,  
 1398, 1470, 2034, 2354, 2361, 2362, 2366,  
 2408, 2419, 2504, 2869, 2886, 3116, 3189, 3232  
 \MT@exp@gcs ..... 302, 306, 310, 312, 315  
 \MT@exp@one@n ..... 317,  
 351, 498, 622, 793, 795, 844, 887, 950,  
 1509, 1937, 2335, 2342, 2372, 2405, 2416,  
 2427, 2446, 2623, 2640, 2669, 2718, 3059, 3907  
 \MT@exp@two@c .....  
 .. 319, 337, 343, 352, 533, 792, 825, 827,  
 829, 838, 1631, 1664, 1667, 2097, 2102, 2185  
 \MT@exp@two@n ..... 321, 905, 914, 2474, 2579  
 \MT@expandfont ..... 1260, 1276, 1280

- \MT@expansion ..... 807, 1248, 3780  
 \MT@expansionfalse ..... 220, 3655, 3745  
 \MT@expansiontrue ..... 220, 3463  
 \MT@extra@context ..... 2427, 2764, 2778,  
     2801, 2819, 2833, 2881, 2884, 2885, 2887,  
     2921, 2936, 2942, 2954, 2959, 3045, 3186,  
     3190, 3193, 3196, 3199, 3203, 3204, 3206, 3233  
 \MT@extra@inputenc ..... 3044, 3068, 3073, 3074  
 \MT@factor@default ..... 253, 3444, 3665  
 \MT@family .... 793, 849, 906, 949, 2007, 2710, 2711  
 \MT@familyalias .....  
     .. 794, 795, 855, 913, 915, 2017, 2019, 2712  
 \MT@feat ..... 859, 982, 1015, 1088, 1090,  
     1092, 1093, 1095, 1096, 1102, 1104, 1107,  
     1115–1119, 1121, 1123–1127, 1130, 1131,  
     1136, 1140, 1143, 1146, 1149–1151, 1160,  
     1161, 1166, 1184, 1633, 1903, 1905, 1908,  
     1909, 1913, 1915, 1918, 2043, 2045, 2047,  
     2051, 2055, 2059, 2060, 2063, 2071, 2073,  
     2074, 2076, 2083, 2087, 2221, 2222, 2251, 2253  
 \MT@features ..... 619,  
     2427, 2440, 2623, 2634, 2663, 2905, 3054, 3488  
 \MT@features@long ..... 619, 622, 627, 2402, 3066  
 \MT@file@list ..... 1924,  
     1926, 1934, 1937, 1939, 1944, 1947, 2718, 2722  
 \MT@find@file ..... 793, 795, 1924  
 \MT@fix@catcode ..... 5  
 \MT@fix@font@set ..... 2493, 3660  
 \MT@font ..... 316,  
     792, 838, 845, 978, 985, 991, 992, 1023,  
     1024, 1029, 1030, 1049, 1050, 1070, 1075,  
     1080, 1174, 1177, 1210, 1211, 1213, 1214,  
     1281, 1285, 1309, 1310, 1350, 1352, 1379,  
     1380, 1386, 1387, 1393, 1394, 1403–1405,  
     1407, 1408, 1413–1415, 1458, 1459, 1465,  
     1466, 1475, 1476, 1478, 1479, 1484, 1485,  
     1509, 1513, 1568, 1877, 2255, 2306–2308,  
     2310, 2323, 2324, 2335, 2337, 2342, 2364, 2373  
 \MT@font@copy ..... 821, 822, 829, 830, 838–840  
 \MT@font@list ..... 2255, 2335–2337, 2396  
 \MT@font@orig ..... 822  
 \MT@font@sets ..... 2478, 2493, 2583, 3660  
 \MT@gdef@n ..... 305, 2757, 2771,  
     2793, 2826, 2840, 2993, 3013, 3026, 3076, 3490  
 \MT@get@axis ..... 2597–2600, 2610  
 \MT@get@basefamily ..... 1936, 1974  
 \MT@get@basefamily@ ..... 1980, 1983  
 \MT@get@char@unit ..... 1014, 1138, 1164, 1252  
 \MT@get@charwd ..... 1048, 1067, 1141, 1164  
 \MT@get@config ..... 3467  
 \MT@get@ex@opt ..... 1251, 1267, 1323  
 \MT@get@ex@opt@ ..... 1331–1333, 1335, 1341  
 \MT@get@font ..... 2578, 2588  
 \MT@get@font@ ..... 2589, 2594, 2893  
 \MT@get@font@and@size ..... 2880, 2892  
 \MT@get@font@dimen ..... 1069, 1144  
 \MT@get@font@dimen@six . 965, 977, 1363, 1442, 1524  
 \MT@get@highlevel ... 2473, 2483, 2612, 2857, 3084  
 \MT@get@inh@list ..... 968, 1254, 1366, 1445, 2069  
 \MT@get@listname ..... 1989, 2045, 2071  
 \MT@get@listname@ ..... 1989  
 \MT@get@ls@basefont ..... 1537, 1654, 1661  
 \MT@get@opt ..... 966, 1113, 1364, 1443  
 \MT@get@range ..... 2502, 2512, 2867  
 \MT@get@size ..... 2519, 2524, 2533, 2552, 2608  
 \MT@get@slot ..... 1012, 1886, 2090, 3108, 3114  
 \MT@get@space@unit .....  
     .. 1138, 1238, 1377, 1384, 1391, 1456, 1463  
 \MT@get@tr@opt ..... 1526, 1602  
 \MT@get@tr@opt@ ..... 1620–1623, 1625  
 \MT@get@unit ..... 1146, 1154, 1613  
 \MT@get@unit@ ..... 1154  
 \MT@getkey ..... 3623, 3638  
 \MT@glet ..... 301, 312, 639,  
     825, 1664, 1684, 1810, 2304, 2397–2400,  
     2712, 2935, 2953, 3129, 3391, 3393, 3837, 3842  
 \MT@glet@nc ..... 311, 1696, 1844, 2083,  
     2324, 2396, 2462, 2464, 2470, 2574, 2786,  
     2979, 2983, 2988, 2991, 2997, 3000, 3003, 3032  
 \MT@glet@nn ..... 314, 2415, 3496  
 \MT@gobble@settings ..... 976, 983  
 \MT@if@false ..... 218,  
     686, 693, 710, 727, 2347, 3231, 3938, 3945  
 \MT@if@list@exists .....  
     ... 964, 1250, 1266, 1362, 1441, 1525, 2042  
 \MT@if@true .....  
     218, 687–689, 694, 695, 711–713, 740, 744,  
     2340, 3240, 3244, 3252, 3257, 3939–3943, 3946  
 \MT@ifdefined@c@T .....  
     .. 325, 690, 716, 1032, 1056, 1311, 1396,  
     1468, 1556, 2172, 2710, 3073, 3176, 3500, 3503  
 \MT@ifdefined@c@TF .....  
     ... 325, 654, 1529, 1540, 1543, 1554,  
     1652, 1721, 1850, 1881, 1882, 2046, 2072, 3185  
 \MT@ifdefined@n@T .....  
     ... 325, 644, 884, 902, 930, 946, 1033,  
     1149, 1185, 1312, 1336, 1397, 1469, 1605,  
     1607, 1626, 2025, 2250, 2460, 2705, 2781,  
     2911, 3135, 3142, 3149, 3155, 3186, 3196, 3623  
 \MT@ifdefined@n@TF ..... 325, 855,  
     865, 885, 903, 931, 947, 1115, 1123, 1325,  
     1342, 1871, 1908, 2013, 2098, 2649, 2652,  
     2678, 3162, 3213, 3222, 3286, 3489, 3493, 3905  
 \MT@ifdim .. 438, 553, 554, 558, 559, 2535, 2544,  
     3238–3240, 3242, 3243, 3250–3252, 3255, 3256  
 \MT@ifdimen .... 403, 2563, 2982, 2990, 3002, 3453  
 \MT@ifempty ..... 358, 991, 992, 1021,  
     1027, 1227, 1228, 1376, 1383, 1390, 1413–  
     1415, 1427–1429, 1431–1433, 1455, 1462,  
     1484, 1485, 1501, 1502, 1702, 1703, 1713,  
     1731, 1820, 1855, 2407, 2439, 2444, 2485,  
     2491, 2513, 2514, 2527, 2528, 2602, 2613,  
     2633, 2638, 2662, 2667, 2740, 2808, 2902,  
     2907, 2921, 2934, 2942, 2952, 2959, 3011,  
     3053, 3058, 3088, 3299, 3330, 3469, 3511, 3568  
 \MT@ifint ..... 371, 2151, 2809, 3012, 3437, 3446  
 \MT@ifstreq ..... 445, 1191, 1904, 2044, 2059,  
     2201, 2204, 2371, 2577, 2879, 2981, 2999,  
     3001, 3177, 3302, 3304, 3307, 3310, 3332,  
     3334, 3348, 3349, 3415, 3416, 3421, 3425,  
     3452, 3513, 3520, 3524, 3531, 3570, 3576, 3879  
 \MT@in@clist ..... 519, 622, 645, 887, 905, 914,  
     950, 1509, 1926, 1937, 2335, 2342, 2405, 2718  
 \MT@in@rlist ..... 547, 933, 2034  
 \MT@in@rlist@ ..... 547  
 \MT@in@rlist@@ ..... 547

- `\MT@in@tlist` ..... 535, 1986, 2416  
`\MT@in@tlist@` ..... 535  
`\MT@in@notfalse` ..... 98  
`\MT@in@nottrue` ..... 98  
`\MT@increment` ..... 583, 3134, 3141, 3148, 3154  
`\MT@info` ..... 75, 90, 95, 3422, 3516, 3522, 3528, 3535, 3573, 3578, 3608, 3612  
`\MT@info@missing@char` ..... 1052, 1078  
`\MT@info@nl` ..... 75, 91, 92, 96, 655, 1079, 3413, 3423, 3476, 3494, 3497, 3658, 3664, 3671, 3752, 3781, 3798, 3804, 3810, 3815, 3822, 3827, 3858, 3864, 3870, 3915, 3926  
`\MT@inh@do` ..... 2082, 3099  
`\MT@inh@feat` ..... 3044, 3053, 3056, 3067  
`\MT@inh@split` ..... 3101, 3105  
`\MT@inlist@false` ..... 519, 523, 536, 548, 2350  
`\MT@inlist@true` ..... 519, 525, 543, 555, 560, 2350  
`\MT@is@active` ..... 2097, 2164  
`\MT@is@char` ..... 2102, 2185, 2191  
`\MT@is@composite` ..... 2100, 2212  
`\MT@is@feature` ..... 621, 2445, 2639, 2668  
`\MT@is@letter` ..... 2095, 2118, 2188, 2217  
`\MT@is@number` ..... 2134, 2139  
`\MT@is@symbol` ..... 2099, 2182  
`\MT@iterate` ..... 569  
`\MT@kerning` ..... 810, 1438, 3826  
`\MT@kerning@false` ..... 226  
`\MT@kerning@true` ..... 226  
`\MT@kn@cc@name` ..... 1447, 1449, 1492, 2833  
`\MT@kn@context` ..... 2390, 2427  
`\MT@kn@doc@contexts` ..... 2427  
`\MT@kn@factor` ..... 229  
`\MT@kn@factor@` ..... 1115  
`\MT@kn@inh@name` ..... 1468–1470  
`\MT@kn@max` ..... 241  
`\MT@kn@min` ..... 241  
`\MT@kn@setname` ..... 2647  
`\MT@kn@split@val` ..... 1453  
`\MT@kn@unit` ..... 229  
`\MT@kn@unit@` ..... 1123, 1495  
`\MT@led@kern` ..... 651  
`\MT@led@unhbox@line` ..... 651  
`\MT@led@mac@setup` ..... 651, 708  
`\MT@let@cn` ..... 313, 314, 315, 856, 972, 1258, 1326, 1370, 1449, 1606, 1608, 1612, 1903, 1915, 2039, 2043, 2076, 2306, 2556, 2915, 3072  
`\MT@let@nc` ..... 311, 650, 1151, 2055, 2087, 2345, 3215, 3220, 3308, 3311, 3527, 3534, 3789  
`\MT@let@nn` ..... 314, 1116, 1121, 1124, 1136, 1343, 1346, 1627  
`\MT@letterspace` ..... 257, 1529, 1606, 1616, 3845, 3846, 3848  
`\MT@letterspace@` ..... 1000, 1056, 1057, 1529–1531, 1534, 1538, 1542, 1550, 1562–1564, 1569, 1579, 1727, 1731, 1819, 1856, 1859  
`\MT@letterspace@default` ..... 257, 3846  
`\MT@listname` ..... 1989, 2036, 2039, 2046, 2047, 2049, 2051, 2072, 2073, 2075, 2076, 2080, 2083, 3116, 3121  
`\MT@load@inputenc` ..... 1192, 1195, 1204  
`\MT@load@list` ..... 970, 1256, 1368, 1447, 1901  
`\MT@loop` ..... 569, 577, 3219  
`\MT@lower` ..... 2506, 2512, 2870  
`\MT@ls@adjust` ..... 1577, 1816, 1817  
`\MT@ls@basefont` ..... 1662–1664, 1667  
`\MT@ls@outer@k` ..... 1580, 1584, 1590, 1744, 1758, 1796, 1864  
`\MT@ls@too@large` ..... 1823, 1827, 3848  
`\MT@ls@font` ..... 1533, 1538, 1550, 1551, 1560, 1563, 1564, 1568, 1631, 1639, 1674, 1676, 1689, 1690, 1697, 1698, 1704, 1708, 1715, 1725, 1729, 1732, 1857, 1860  
`\MT@ls@lig` ..... 1651  
`\MT@lua` ..... 283, 375, 407, 449  
`\MT@map@c@list@` ..... 491  
`\MT@map@c@list@c` ..... 491, 831, 1885, 2341, 2353, 2360, 2395, 2402, 2440, 2442, 2634, 2636, 2663, 2665, 2905, 3054, 3056, 3066, 3488, 3621  
`\MT@map@c@list@n` ..... 491, 864, 1870, 2427, 2471, 2500, 2575, 2623, 2695, 2755, 2855, 2865, 2877, 2995, 3009, 3111, 3294, 3297, 3326, 3328, 3357, 3432, 3509, 3566  
`\MT@map@tlist@` ..... 509  
`\MT@map@tlist@c` ..... 509, 538, 549, 834, 1034, 1313, 1398, 1470, 1993, 2366, 3187, 3232, 3660  
`\MT@map@tlist@n` ..... 509, 1965, 2193, 3128  
`\MT@maybe@do` ..... 859, 962, 1248, 1360, 1439, 1511  
`\MT@maybe@etex` ..... 262  
`\MT@maybe@rem@from@list` ..... 2366, 2370  
`\MT@MT` ..... 2, 75, 78, 81, 83, 85, 90, 91, 93, 94, 202, 209, 290, 633, 703, 705, 2944, 3470, 3608, 3612, 3659, 3883, 3919, 3920  
`\MT@next@listname` ..... 1996, 2005  
`\MT@next@listname@` ..... 2014, 2026, 2032  
`\MT@nl@ligatures` ..... 1877, 2734  
`\MT@nl@setname` ..... 1869, 2734  
`\MT@noligatures` ..... 812, 1866, 3894  
`\MT@noligatures@` ..... 1674, 1676, 1877, 1880  
`\MT@noligatures@false` ..... 223  
`\MT@noligatures@true` ..... 223, 2742  
`\MT@nonselected@false` ..... 1263, 1277  
`\MT@nonselected@true` ..... 1263, 1265  
`\MT@noreset@false` ..... 2117, 2124, 2133, 2154, 2201, 2204  
`\MT@noreset@true` ..... 2093, 2117, 2142  
`\MT@old@cmd` ..... 65, 70–73  
`\MT@opt@autofalse` ..... 3271  
`\MT@opt@autotru` ..... 3271, 3359  
`\MT@opt@def@set` ..... 3285, 3313, 3338  
`\MT@opt@DVIfalse` ..... 3272  
`\MT@opt@DVIttrue` ..... 3272, 3362, 3365  
`\MT@opt@expansion@false` ..... 3270  
`\MT@opt@expansion@true` ..... 3270  
`\MT@options` ..... 2843  
`\MT@optwarn@admissible` ..... 3273, 3350, 3425  
`\MT@optwarn@nan` ..... 3277, 3439, 3448  
`\MT@orig@add@accent` ..... 2325  
`\MT@orig@foreign@language` ..... 3932, 3934  
`\MT@orig@pickup@font` ..... 733, 774, 779, 2260, 2327  
`\MT@orig@py@macron` ..... 771, 775, 780  
`\MT@orig@select@language` ..... 3927, 3929  
`\MT@outer@kern` ..... 1579, 1582, 1588, 1590, 1591, 1596, 1655, 1656, 1658, 1796, 1838, 1864  
`\MT@outer@space` ..... 1572, 1573, 1687, 1740, 1742, 1743, 1756, 1757, 1768, 1769, 1784, 1785, 1789, 1790  
`\MT@pdf@annot` ..... 98

- \MT@pdfTeX@no ..... 146, 196, 200, 275  
 \MT@permute 2770, 2792, 2806, 2825, 2839, 3079, 3125  
 \MT@permute@ ..... 3125  
 \MT@permute@@ ..... 3125  
 \MT@permute@@@ ..... 3125  
 \MT@permute@@@ ..... 3125  
 \MT@permute@@@@ ..... 3153, 3158  
 \MT@permute@@@@@ ..... 3166, 3168  
 \MT@permute@define ..... 3159, 3169–3171, 3211  
 \MT@permute@reset ..... 3128, 3217  
 \MT@permutelist .....  
     ..... 2764, 2778, 2804, 2819, 2833, 2881,  
     2882, 2887, 3077, 3181, 3182, 3186, 3190,  
     3193, 3196–3199, 3206, 3207, 3233, 3264, 3265  
 \MT@pickupfont ..... 776, 781, 2321, 2329  
 \MT@plain ..... 123, 143  
 \MT@pr@c@name ..... 970, 972, 1218, 2764  
 \MT@pr@context ..... 821, 2389, 2427  
 \MT@pr@doc@contexts ..... 2427  
 \MT@pr@factor ..... 229, 3447, 3665, 3666  
 \MT@pr@factor@ ..... 1115  
 \MT@pr@inh@name ..... 1032, 1033, 1035  
 \MT@pr@level ..... 229, 3663, 3664  
 \MT@pr@max ..... 241  
 \MT@pr@min ..... 241  
 \MT@pr@setname ..... 2647  
 \MT@pr@split@val ..... 1019  
 \MT@pr@unit ..... 229, 3454, 3667  
 \MT@pr@unit@ ..... 1123, 1221  
 \MT@preset@aux ..... 1223,  
     1225, 1227, 1228, 1231, 1497, 1499, 1501, 1502  
 \MT@preset@aux@factor 1223, 1231, 1427–1429, 1497  
 \MT@preset@aux@space . 1225, 1231, 1431–1433, 1499  
 \MT@preset@ex ..... 1337, 1354  
 \MT@preset@kn ..... 1490  
 \MT@preset@kn@ ..... 1490  
 \MT@preset@pr ..... 1216  
 \MT@preset@pr@ ..... 1216  
 \MT@preset@sp ..... 1420  
 \MT@preset@sp@ ..... 1420  
 \MT@ProcessOptionsWithKV ..... 3616, 3639  
 \MT@protrusion ..... 806, 962, 3670  
 \MT@protrusionfalse ..... 219  
 \MT@protrusiontrue ..... 219, 3460  
 \MT@rbba@expansion ..... 614  
 \MT@rbba@kerning ..... 614  
 \MT@rbba@protrusion ..... 614  
 \MT@rbba@spacing ..... 614  
 \MT@rbba@tracking ..... 614  
 \MT@register@font ..... 815, 2337, 2398  
 \MT@register@font@cx ..... 2359, 2398  
 \MT@register@subst@font ..... 2312, 2336, 2399  
 \MT@register@subst@font@cx ..... 2352, 2399  
 \MT@rem@from@clist ..... 530, 844, 2372, 2847  
 \MT@rem@from@list ..... 834, 842  
 \MT@rem@last@space ..... 346  
 \MT@repeat ..... 569, 579, 3225  
 \MT@requires@latex ..... 142, 191, 636,  
     729, 758, 1630, 2258, 3278, 3431, 3616, 3620  
 \MT@requires@luatex ..... 278, 373,  
     405, 447, 2925, 3380, 3563, 3796, 3807, 3818  
 \MT@requires@pdfTeX ..... 274,  
     371, 403, 445, 651, 789, 798, 809, 811,  
     819, 998, 1040, 1054, 1288, 1359, 1438,  
     1506, 1671, 1866, 2734, 2923, 3025, 3031,  
     3378, 3462, 3562, 3706, 3731, 3784, 3874, 3891  
 \MT@res@a ..... 468, 470, 521, 528, 531, 533, 537, 542  
 \MT@res@b ..... 469, 470, 531–533, 541, 542  
 \MT@reset@context ..... 2380, 2383, 2387, 2414  
 \MT@reset@context@ ..... 2387, 2414  
 \MT@reset@ef@codes ..... 1253, 1275, 1288, 1338  
 \MT@reset@ef@codes@ ..... 1283, 1291, 1295  
 \MT@reset@kn@codes ..... 1444, 1451, 1481  
 \MT@reset@kn@codes@ ..... 1481  
 \MT@reset@pr@codes ..... 967, 974, 995  
 \MT@reset@pr@codes@ ..... 995  
 \MT@reset@sp@codes ..... 1365, 1372, 1410  
 \MT@reset@sp@codes@ ..... 1410  
 \MT@restore@catcodes ... 5, 7, 8, 213, 298, 634, 3960  
 \MT@restore@p@h ..... 684, 696, 745  
 \MT@saved@setupfont ..... 3606, 3613  
 \MT@scale ..... 590, 1043, 1089, 1103, 1106, 1300  
 \MT@scale@factor .... 1045, 1064, 1087, 1233, 1356  
 \MT@scale@to@em .. 1022, 1028, 1040, 1061, 1239,  
     1378, 1385, 1392, 1457, 1464, 1615, 1722, 1851  
 \MT@selectedfalse ..... 222  
 \MT@selectedtrue ..... 222  
 \MT@series ..... 849, 949, 2008, 2020  
 \MT@set@all@ex ..... 1283, 1357  
 \MT@set@all@kn ..... 1481, 1503  
 \MT@set@all@pr ..... 988, 995, 1229  
 \MT@set@all@sp ..... 1410, 1435  
 \MT@set@babel@context .... 3904, 3930, 3935, 3949  
 \MT@set@codes .. 973, 1003, 1259, 1371, 1450, 1916  
 \MT@set@curr@ls ..... 1569  
 \MT@set@curr@ok ..... 1582, 1583, 1596, 1773  
 \MT@set@curr@os ..... 1571, 1774  
 \MT@set@default@set ..... 2663, 2669, 2676  
 \MT@set@ex@codes ..... 1279, 3736, 3738  
 \MT@set@ex@codes@n ..... 1263, 1279, 3738  
 \MT@set@ex@codes@rs ..... 1249, 3736  
 \MT@set@ex@heirs ..... 1313, 1349  
 \MT@set@inh@list ..... 3050, 3052  
 \MT@set@inputenc 969, 1182, 1255, 1367, 1446, 2081  
 \MT@set@inputenc@ ..... 1185, 1187  
 \MT@set@kn@codes ..... 1440  
 \MT@set@kn@heirs ..... 1470, 1474  
 \MT@set@listname .....  
     971, 1114, 1257, 1324, 1369, 1448, 1604, 2220  
 \MT@set@lsfont ..... 1567, 1630, 1639  
 \MT@set@named@keys 2766, 2780, 2802, 2821, 2835, 2843  
 \MT@set@pr@codes ..... 963  
 \MT@set@pr@heirs ..... 1036, 1209  
 \MT@set@sp@codes ..... 1361  
 \MT@set@sp@heirs ..... 1398, 1402  
 \MT@set@tr@codes ..... 1521, 1634, 1644  
 \MT@SetExpansion ..... 2776, 2778  
 \MT@SetExtraKerning ..... 2831, 2833  
 \MT@SetExtraSpacing ..... 2817, 2819  
 \MT@SetProtrusion ..... 2762, 2764  
 \MT@SetTracking ..... 2798, 2800  
 \MT@setup@ ..... 635, 638, 639, 3372, 3373  
 \MT@setup@contexts ..... 2379, 2394  
 \MT@setupfont .. 788, 2315, 3606, 3609, 3613, 3647  
 \MT@setupfont@hook 685, 709, 715, 720, 745, 752, 788  
 \MT@shape ..... 849, 949, 2009, 2021



- \MT@shorthandoff ..... 3913, 3944, 3947  
 \MT@showpdfannot ..... 104  
 \MT@shrink ..... 229, 1270,  
     3677, 3678, 3681, 3682, 3685, 3689, 3741, 3754  
 \MT@shrink@ ..... 1270, 1281, 1323  
 \MT@shrink@default ..... 254  
 \MT@size ..... 554, 558, 559, 849, 949  
 \MT@size@name ..... 547, 2036  
 \MT@sp@cc@name ..... 1368, 1370, 1422, 2819  
 \MT@sp@context ..... 2390, 2427, 3879  
 \MT@sp@doc@contexts ..... 2427  
 \MT@sp@factor ..... 229  
 \MT@sp@factor@ ..... 1115  
 \MT@sp@inh@name ..... 1396–1398  
 \MT@sp@max ..... 241  
 \MT@sp@min ..... 241  
 \MT@sp@setname ..... 2647  
 \MT@sp@split@val ..... 1374  
 \MT@sp@unit ..... 229  
 \MT@sp@unit@ ..... 1123, 1425  
 \MT@spacing ..... 810, 1359, 3814  
 \MT@spacingfalse ..... 225  
 \MT@spacingtrue ..... 225  
 \MT@split@codes ..... 1005, 1009  
 \MT@split@name ..... 792, 849  
 \MT@step ..... 229,  
     1271, 3680, 3696, 3703, 3755, 3760, 3761, 3765  
 \MT@step@ ..... 1271, 1281, 1323  
 \MT@step@default ..... 254  
 \MT@stretch ..... 229, 1269, 3674,  
     3675, 3678, 3681, 3683, 3688, 3691, 3740, 3754  
 \MT@stretch@ ..... 1269, 1281, 1323  
 \MT@stretch@default ..... 254, 3675  
 \MT@temp ..... 990–993,  
     1412–1416, 1483–1486, 1692, 1696, 1699,  
     1704, 1707, 1712, 1768, 1775, 1778, 1781,  
     1784, 1789, 1796, 1799, 1841, 1844, 1861,  
     2596, 2621, 3619, 3624, 3629, 3630, 3632, 3635  
 \MT@tempencoding ..... 3161, 3162, 3172, 3178  
 \MT@tempfamily ..... 3173  
 \MT@tempseries ..... 3174  
 \MT@tempshape ..... 3175  
 \MT@tempsize ..... 3129, 3176, 3185, 3187, 3191  
 \MT@test@ast ..... 1720, 1849, 2484, 2489  
 \MT@textls ..... 49, 1816, 1817, 1819  
 \MT@the@pr@code ..... 997, 1023, 1029, 1565  
 \MT@the@pr@code@tr ..... 997, 1565  
 \MT@tlist@break ..... 509, 544, 566, 1998, 3267  
 \MT@toks ..... 217, 1079, 1109, 1319,  
     2094, 2122, 2129, 2176, 2206, 2225, 2237, 2245  
 \MT@tr@cc@name .....  
     ... 1605–1608, 1612, 1626, 1627, 2803, 2810  
 \MT@tr@context ..... 2390, 2427  
 \MT@tr@doc@contexts ..... 2427  
 \MT@tr@factor@ ..... 1614  
 \MT@tr@font@list ..... 1507  
 \MT@tr@ispace ..... 1540, 1541, 1620  
 \MT@tr@ligatures ..... 1556, 1623, 1673, 1676  
 \MT@tr@max ..... 241, 1828–1830  
 \MT@tr@min ..... 241, 1832–1834  
 \MT@tr@noligatures ..... 1556, 1670  
 \MT@tr@okern ..... 1554, 1622  
 \MT@tr@ospace ..... 1543, 1544, 1620  
 \MT@tr@outer@icr ..... 1778, 1801  
 \MT@tr@outer@icr@ ..... 1801  
 \MT@tr@outer@l ..... 1575, 1736  
 \MT@tr@outer@next ..... 1764, 1803  
 \MT@tr@outer@r ..... 1598, 1764, 1801  
 \MT@tr@outer@re ..... 1764  
 \MT@tr@set@okern ..... 1555, 1838  
 \MT@tr@set@okern@ ..... 1842, 1843, 1848  
 \MT@tr@set@space ..... 1546, 1688  
 \MT@tr@set@space@ ..... 1693–1695, 1701  
 \MT@tr@set@space@@ ..... 1706, 1711, 1714, 1719  
 \MT@tr@setname ..... 2647  
 \MT@tr@unit@ ..... 1607, 1721, 1850  
 \MT@tracking 799, 1507, 1634, 2317, 3590, 3591, 3803  
 \MT@tracking@ ..... 1507, 3591  
 \MT@trackingfalse ..... 227  
 \MT@trackingtrue ..... 227  
 \MT@try@order ..... 1993, 2001  
 \MT@undefined@char ..... 2169, 2171, 2181  
 \MT@upper ..... 2506, 2512, 2870  
 \MT@use@set ..... 2634, 2640, 2647  
 \MT@val ..... 2407–2409, 2411, 2417, 2419,  
     2422, 2472, 2476, 2484, 2486, 2501–2503,  
     2515, 2518, 2520, 2523, 2525, 2526, 2532,  
     2534, 2535, 2537, 2540, 2542, 2553, 2556,  
     2560, 2563–2565, 2567, 2569, 2576–2578,  
     2581, 2590–2592, 2604, 2606, 2611, 2613,  
     2615, 2617, 2856, 2858, 2866–2868, 2878–  
     2880, 2883, 2888, 3082, 3083, 3085, 3089,  
     3110, 3116, 3120, 3121, 3286, 3287, 3290,  
     3298, 3299, 3302, 3304, 3307, 3310, 3329,  
     3330, 3332, 3334, 3510, 3511, 3513, 3520,  
     3524, 3531, 3537, 3567, 3568, 3570, 3576, 3579  
 \MT@variants ..... 1986, 2688, 2698  
 \MT@vinfo ..... 75, 92, 805, 877, 1118, 1127,  
     1131, 1166, 1327, 1344, 1523, 1531, 1888,  
     1897, 1909, 1933, 1943, 1946, 2049, 2051,  
     2057, 2388, 2411, 2720, 2726, 3413, 3420, 3906  
 \MT@warn@ascii ..... 2131, 2224  
 \MT@warn@axis@empty ..... 2603, 2614, 2619  
 \MT@warn@code@too@large ..... 1093, 1096, 1100  
 \MT@warn@err ..... 75, 3417, 3418  
 \MT@warn@ex@too@large ..... 1303, 1306, 1317  
 \MT@warn@lua ..... 3786, 3796, 3807, 3818  
 \MT@warn@maybe@inputenc ... 1883, 2238, 2246, 2249  
 \MT@warn@nodim ..... 2974, 2984, 2992, 3004  
 \MT@warn@number@too@large ..... 2158, 2230  
 \MT@warn@preset@tewidth ... 1222, 1242, 1426, 1496  
 \MT@warn@rest ..... 2111, 2235  
 \MT@warn@tracking@DVI ..... 1532, 3830  
 \MT@warn@unknown ..... 2108, 2243  
 \MT@warning ..... 66, 75, 93, 1829,  
     1833, 1891, 2060, 2461, 2567, 2620, 2706,  
     2728, 2811, 2912, 2975, 3015, 3028, 3090,  
     3179, 3197, 3264, 3417, 3422, 3500, 3598–3600  
 \MT@warning@nl .....  
     . 75, 87, 94, 199, 289, 632, 667, 676, 722,  
     979, 1071, 1107, 1159, 1199, 1243, 1318,  
     1681, 2225, 2231, 2236, 2244, 2288, 2783,  
     2961, 3274, 3280, 3290, 3418, 3423, 3455,  
     3483, 3644, 3698, 3742, 3763, 3832, 3880, 3951  
 \MT@while@num ..... 575, 581, 748–750

- `\MT@with@babel@and@T` ..... 643, 687–689, 711–713, 3939–3943, 3946  
`\MT@with@package@T` ..... 641, 691, 694, 695, 701, 708, 718, 730, 744, 746, 759, 770, 2299  
`\MT@xadd` ..... 102, 477, 1934, 1939, 1944, 1947, 2354, 2362, 2419, 2504, 2722, 2869, 3116  
`\MT@xaddb` ..... 484, 2886, 3189  
`\MT@xdef@n` .. 307, 2650, 2653, 2680, 2685, 2709, 2810, 2881, 2903, 3040, 3074, 3206, 3287, 3289
- N**
- `\normalfont` ..... 787, 3773, 3777  
`\nullfont` ..... 787
- P**
- `\pdfadjustinterwordglue` ..... 3592, 3809  
`\pdfadjustspacing` ..... 3561, 3751, 3776  
`\pdfappendkern` ..... 3594, 3821  
`\pdfcopyfont` ..... 829  
`\pdfcprot` (package) ..... 4, 27, 29, 50, 59, 87, 180  
`\pdffontexpand` ..... 1281  
`\pdfnoligatures` ..... 1890, 1896  
`\pdfprependkern` ..... 3593, 3820  
`\pdfprotrudechars` ..... 3560, 3663, 3800, 3841  
`\pdftracingfonts` ..... 277  
`\pickup@font` ..... 733, 774, 776, 779, 781, 2287, 2289, 2298, 2321, 2327, 2329  
`pifont` (package) ..... 87  
`pinyin` (package) ..... 52, 185  
`protrusion` (option) ..... 6, 7, 11, 111, 112  
`PSNFSS` (package) ..... 176  
`pstricks` (package) ..... 9, 120  
`pxfonts` (package) ..... 21, 128
- Q**
- `qfonts` (package) ..... 21, 128, 183
- R**
- `ragged2e` (package) ..... 74, 186  
`relsize` (package) ..... 95  
`\rightmarginkern` ..... 660  
`\rpxcode` ..... 992, 1029, 1030, 1211, 1214, 1564
- S**
- `selected` (option) ..... 8, 14, 32, 64, 81, 112, 181  
`\selectfont` ..... 1635, 2382, 2392, 3957  
`\set@fontsize` ..... 2560  
`\SetExpansion` ..... 14, 36, 2774  
`\SetExtraKerning` ..... 19, 38, 2829  
`\SetExtraSpacing` ..... 18, 39, 2815  
`\SetProtrusion` ..... 13, 35, 2760
- `\SetTracking` ..... 16, 37, 2796  
`\sfcode` ..... 3878  
`\shbcode` ..... 1393, 1394, 1405, 1408, 1415  
`\showhyphens` ..... 3771, 3775  
`shrink` (option) .. 8, 16, 32, 114, 120, 122, 183, 185  
`slantsc` (package) ..... 154  
`soul` (package) ..... 4, 29, 51, 52, 75, 184, 186  
`soulutf8` (package) ..... 29, 186  
`\spacefactor` ..... 1747  
`\spaceskip` ..... 1748, 1751  
`spacing` (option) ..... 7, 11, 26, 30, 112  
`\stbcode` ..... 1386, 1387, 1404, 1408, 1414  
`step` (option) ... 8, 32, 114, 120, 122, 181, 183, 185  
`stretch` (option) ... 8, 16, 25, 32, 114, 120, 122, 185
- T**
- `\tagcode` ..... 1560, 1882, 1887  
`tex4ht` (package) ..... 50, 184  
`\textls` .. 24, 52, 53, 738, 761, 766, 1578, 1647, 1814  
`\textls*` ..... 24  
`\textmicrotypecontext` ..... 22, 48, 734, 2386  
`tipa` (package) ..... 87  
`trace` (package) ..... 89, 184  
`\tracingmicrotype` ..... 86  
`\tracingmicrotypeinpdf` ..... 97  
`\tracingmicrotypeinpdfallfalse` ..... 103  
`\tracingmicrotypeinpdfalltrue` ..... 103  
`tracking` (option) ..... 7, 11, 24, 30, 112  
`txfonts` (package) ..... 21, 128
- U**
- `ucs` (package) ..... 84, 186  
`ulgothic` (package) ..... 21  
`unit` (option) ..... 8, 31, 182–184  
`\UseMicrotypeSet` ..... 11, 33, 60, 72, 2455, 2631
- V**
- `verbose` (option) ..... 9, 30, 32, 114, 181, 182, 186
- W**
- `wordcount` (package) ..... 48, 185
- X**
- `\xspaceskip` ..... 1750, 1753
- Y**
- `yfonts` (package) ..... 24
- Z**
- `zefonts` (package) ..... 21, 127

## C The L<sup>A</sup>T<sub>E</sub>X Project Public License

*LPPL Version 1.3c 2006-05-20*

Copyright 1999, 2002–2006 L<sup>A</sup>T<sub>E</sub>X3 Project

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

### Preamble

The L<sup>A</sup>T<sub>E</sub>X Project Public License (LPPL) is the primary license under which the 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.

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 ‘modguide.tex’ in the base L<sup>A</sup>T<sub>E</sub>X distribution for suggestions.

### Definitions

In this license document the following terms are used:

**Work:** Any work being distributed under this License.

**Derived Work:** Any work that under any applicable law is derived from the Work.

**Modification:** Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

**Modify:** To apply any procedure that produces a Derived Work under any applicable law.

**Distribution:** Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun’s Network File System (NFS).

**Compiled Work:** A version of the Work that has been processed into a form where it is directly usable on a

computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

**Current Maintainer:** A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the ‘Copyright Holder’ under any applicable law.

**Base Interpreter:** A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a ‘L<sup>A</sup>T<sub>E</sub>X-Format’ or in the case of files belonging to the ‘L<sup>A</sup>T<sub>E</sub>X-format’ a program implementing the ‘T<sub>E</sub>X language’.

### Conditions on Distribution and Modification

- Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.
- You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
- You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
- If you are the Current Maintainer of the Work, you

may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.

5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
  - (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
  - (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.
  - (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
  - (d) You distribute at least one of the following with the Derived Work:
    - i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
    - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.
10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
  - (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.
11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

## No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by

anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder

or said author or said other party has been advised of the possibility of such damages.

## Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
  - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
  - (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current

Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a 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.