

# The microtype package

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

R Schlicht  
w.m.l@gmx.net

v2.3  
2007/12/23

---

## Abstract

The `microtype` package provides a  $\LaTeX$  interface to the micro-typographic extensions of pdf $\TeX$ : 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 $\TeX$ , at least version 0.14f. Automatic font expansion requires version 1.20 or newer. Disabling ligatures requires pdf $\TeX$  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.

The alternative package `letterspace`, which also works with `eplain` or even `minilts`, provides the user commands for letterspacing only, omitting support for all other extensions (see section 7).

This package is copyright © 2004–2007 R Schlicht. It may be distributed and/or modified under the conditions of the  [\$\LaTeX\$  Project Public License](#), either version 1.3c of this license or (at your option) any later version. This work has the LPPL maintenance status ‘author-maintained’.

---

<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 2 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	Micro-Typographic Options . . . . .	6
3.2	Options for Character Protrusion . . . . .	7
3.3	Options for Font Expansion . . . . .	7
3.4	Option for Tracking/Letterspacing . . . . .	8
3.5	Miscellaneous Options . . . . .	8
3.6	Changing Options Later . . . . .	9
<b>4</b>	<b>Selecting Fonts for Micro-Typography</b>	<b>10</b>
<b>5</b>	<b>Micro Fine Tuning</b>	<b>12</b>
5.1	Character Protrusion . . . . .	12
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>23</b>
<b>8</b>	<b>Disabling Ligatures</b>	<b>24</b>
<b>9</b>	<b>Hints and Caveats</b>	<b>25</b>
<b>10</b>	<b>Contributions</b>	<b>27</b>
<b>11</b>	<b>Acknowledgments</b>	<b>28</b>
<b>12</b>	<b>References</b>	<b>28</b>
<b>13</b>	<b>Short History</b>	<b>29</b>
<b>14</b>	<b>Implementation</b>	<b>32</b>
14.1	Preliminaries . . . . .	33
	[Debugging – 34] [Requirements – 35] [Declarations – 38] [Auxiliary Macros – 39] [Compatibility – 47]	
14.2	Font Setup . . . . .	51
	[Protrusion – 56] [Expansion – 62] [Interword Spacing (Glue) – 65] [Additional Kerning – 67] [Tracking – 69] [Disabling Ligatures – 76] [Loading the Configuration – 77] [Translating Characters into Slots – 81] [Hook into $\LaTeX$ 's font selection – 86] [Context-sensitive Setup – 89]	

14.3	Configuration . . . . .	91
	[Font Sets – 91] [Variants and Aliases – 97] [Disabling Ligatures – 98]	
	[Interaction with babel – 98] [Fine Tuning – 98] [Character	
	Inheritance – 104] [Permutation – 106]	
14.4	Package Options . . . . .	109
	[Declaring the Options – 109] [Reading the Configuration File – 113]	
	[Hook for Other Packages – 114] [Changing Options Later – 115]	
	[Processing the Options – 117]	
<b>15</b>	<b>Configuration Files</b>	<b>124</b>
15.1	Font Sets . . . . .	124
15.2	Font Variants and Aliases . . . . .	125
15.3	Interaction with babel . . . . .	126
15.4	Note on Admissible Characters . . . . .	127
15.5	Character Inheritance . . . . .	127
	[OT1 – 127] [T1 – 128] [LY1 – 128] [OT4 – 129] [QX – 130] [T5 – 130]	
	[Euro symbols – 131]	
15.6	Tracking . . . . .	131
15.7	Font Expansion . . . . .	132
15.8	Character Protrusion . . . . .	133
	[Default – 133] [Italics – 141] [Small Caps – 149] [Italic Small Caps – 152]	
	[textcomp – 154] [Computer Modern math – 159] [AMS symbols – 162]	
	[Euler – 166] [Euro symbols – 170]	
15.9	Interword Spacing . . . . .	171
	[Nonfrenchspacing – 173]	
15.10	Additional Kerning . . . . .	174
	[French – 174] [Turkish – 175]	
<b>16</b>	<b>Auxiliary File for Micro Fine Tuning</b>	<b>175</b>
<b>A</b>	<b>Change History</b>	<b>178</b>
<b>B</b>	<b>Index</b>	<b>184</b>
<b>C</b>	<b>The L<sup>A</sup>T<sub>E</sub>X Project Public License</b>	<b>192</b>

## List of Tables

1	Predefined font sets . . . . .	11
2	Fonts with tailored protrusion settings . . . . .	21
3	Order for matching font attributes . . . . .	79

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

pdf<sub>T</sub><sub>E</sub>X, the <sub>T</sub><sub>E</sub>X 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 <sub>T</sub><sub>E</sub>X and  $\text{\LaTeX}$  users: font expansion can now take place automatically. That is, pdf<sub>T</sub><sub>E</sub>X 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<sub>T</sub><sub>E</sub>X 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 <sub>T</sub><sub>E</sub>X. 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<sub>T</sub><sub>E</sub>X's new extension, this doesn't pose a problem any longer. 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 `\lstyle` 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 words ends with an ‘r’, the following space should be a tiny bit smaller than that following an, say, ‘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 (and 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 pdfT<sub>E</sub>X version and/or the output mode).

### 3.1 Micro-Typographic Options

**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 pdfT<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 pdfT<sub>E</sub>X):

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

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

```
\usepackage{microtype}
```

When pdfTeX employs font expansion and character protrusion, line breaks (and consequently, page breaks) may turn out differently. If that is not desired, you may pass the value `compatibility` to the protrusion and/or expansion options. Typographically, however, the results may be suboptimal, hence the default value is `nocompatibility`.

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

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

**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.

**kerning**

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

### 3.2 Options for Character Protrusion

**factor** *<integer>* 1000

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

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

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

### 3.3 Options for 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 *<font name>±<expansion value>*, 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`).

<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'.

**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.

Beginning with version 1.5, where this option was introduced, it is by default set to false, so that all characters will be expanded by the same amount. See section 5.2 for a more detailed discussion.

### 3.4 Option for Tracking/Letterspacing

**letterspace**  $\langle integer \rangle$  100

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

### 3.5 Miscellaneous Options

**DVIoutput** true, false \* false

pdfTeX 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

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



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.

<b>draft</b>	true, false	false
<b>final</b>	If the draft option is passed to the package, <i>all micro-typographic extensions will be disabled</i> , which may lead to different line, and hence page, breaks. The 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.	
<b>verbose</b>	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, which is disabled by default. 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.	
<b>babel</b>	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.	
<b>config</b>	<i>&lt;file name&gt;</i>	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 <i>without the extension</i> , e. g., config=mycrotype.	

### 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 the keys: **expansion**, **protrusion** and **activate**, which in turn may receive the values true, false, compatibility or nocompatibility (but not the name of a font set). Since there is no compatibility level for **tracking**, **spacing** and **kerning**, only the values true and false are admissible here. 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 is that the encoding must match.

If a value is followed by an asterisk (like ‘`rm*`’ and ‘`sf*`’ in the above example), it does not designate an NFSS code, but will expand to the document’s `\<value>default`, e.g., `\rmdefault`. A single asterisk means `\<attribute>default`, e.g., `\encodingdefault`, respectively `\normalsize` for the size axis. Sizes may either be specified as a dimension (‘10’ or ‘10pt’), or as a size selection command *without* the backslash. You may also specify ranges (e.g., ‘`small-Large`’); while the lower boundary is included in the range, the upper boundary is not. Thus, ‘12-16’ would

Table 1: Predefined font sets

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

match 12pt, 13.5pt and 15.999pt, e. g., but not 16pt. You are allowed to omit the lower or upper bound ('-10', 'large-').

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

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

As you can tell from the example, the asterisk notation is also 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 1 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 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.

### 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, } }
```

<sup>4</sup> For the interested, table 3 on page 79 presents the exact order.

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 Computer Modern Roman family in encoding T1.

*The protrusion settings* consist of  $\langle \text{character} \rangle = \langle \text{protrusion factors} \rangle$  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. You also have the possibility to declare lists of characters that should inherit protrusion or expansion factors (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 1em 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** Select 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. 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.

<sup>5</sup> The `unit` option can even be passed globally to the package. 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.

*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 (contexts are only possible if you are using at least pdf $\TeX$  version 1.40.4).

**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 pdf $\TeX$  version 1.40.4 or later (for older versions, a dirty trick is laid out in section 14.2 on page 53). Also note that pdf $\TeX$  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  $\text{\TeX}$  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 1em (or the given unit); negative values are allowed, too.

*Options:*

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

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

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

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

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

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

**no ligatures** As far as pdf $\text{\TeX}$  is concerned, ligatures in letterspaced fonts would be constructed as usual, which may be advisable when changing the tracking 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.



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; 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 0.45em) immediately before the piece of text may *stretch* by 0.25em and *shrink* by 0.15em. Note that there is no outer space after the text, since the exclamation mark immediately follows; instead, the default *outer kern* of half the letterspace amount (0.08em) is added. Furthermore, one *ligature* wasn’t broken up, because we neglected to specify the ‘s’ in the `no ligatures` key.

As another example, suppose you want to space out all small capitals by 50/1000em, fonts smaller than `\small` by 0.02em, and to decrease the tracking of large type by 0.02em. You can achieve this with the following settings:

<sup>7</sup> The inseparable connexion of ligatures and kerns is a limitation of  $\text{T}_{\text{E}}\text{X}$  that will not be lifted before the advent of  $\text{LuaT}_{\text{E}}\text{X}$ .

<sup>8</sup> With  $\text{pdfT}_{\text{E}}\text{X}$  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.

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

Letterspaced fonts for which settings don't exist will be spaced out by the default of 0.1em (adjustable with the package option `letterspace`, see section 3.5). Suppose 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 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 well as the maximum stretch and the maximum shrink amount of the interword space ( $\text{\fontdimen 3}$  and  $4$ ). As another example, setting all three value to  $-1000$  would completely cancel a space after the respective character.

## 5.5 Additional Kerning

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

Fine tune the additional 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 additional kerning relates to single characters, that is, whenever that 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 pdf $\text{\TeX}$  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 are relative to 1em.

**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, the following will happen : a thin space will be inserted in front of each semicolon, exclamation and question mark ; a normal space in front of the colon. Read section 6 to learn how to activate these settings !

## 5.6 Character Inheritance

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

In most cases, accented characters should inherit the settings from the respective base character. For example, all of the characters À, Á, Â, Ã, Ä, Å and Æ should probably be protruded by the same (absolute) amount as the character A. Using the command `\DeclareCharacterInheritance`, you may declare such classes of characters, so that you then only have to set up the respective base character. With the optional argument, which may contain a comma-separated list of features, you can confine the scope of the list. Additionally, it accepts the `inputenc` key to set the input encoding for this list. The font set can be declared in the usual way, 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 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 2 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. [Fontname](#)). This allows it to put settings for, e. g., the fonts `padx` (expert set),

Table 2: 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 (qp1), 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)

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 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 clones of 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. 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.

`\textmicrotypecontext` {⟨context assignments⟩} {⟨general text⟩}

The alternative command `\textmicrotypecontext` sets the context(s) for the text given in the second argument.

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

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

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

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

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

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

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:

<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.

```
\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. The current language will then be automatically detected and the contexts set accordingly.

```
\DeclareMicrotypeBabelHook {⟨list of babel languages⟩} {⟨context list⟩}
```

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

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

Consequently, whenever you switch to the French language, the kerning context will be changed to ‘french’ and the spacing context will be reset. This hook only has an effect if the package was loaded with the `babel` option (see section 3.5). 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  $\text{\LaTeX}$ ’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  $100/1000\text{em} = 0.1\text{em}$ ; this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option, with decreasing significance in this order.

<sup>10</sup> Letterspacing should be used cautiously; in particular, letterspacing 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.

`\lslig` {*<ligature>*}

Since the commands `\textls` and `\lsstyle` will also evaluate the ‘no ligatures’ key for the respective font, you do not need to 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 typesetting text in Fraktur, 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, ‘`Uusſiđtſloſigkeit`’).

```
\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 `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). If you prefer to forgo `microtype`’s specialties, you may load the `letterspace` package instead. Both packages should not be used at the same time.

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

## 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.

## 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 a large factor (depending on expansion limits and step). Therefore, courtesy and thriftiness of bandwidth command it not to enable font expansion when creating files to be distributed electronically. With pdfTeX 1.40, 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 1) 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, you would also have to create settings for the fonts in question (see section 5.1). It goes without saying that contributions for these encodings are more than welcome.

*Only employ kerning adjustment if it is customary in the language's typographic tradition.* In contrast to protrusion and expansion (and possibly adjustment of interword spacing), 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:<sup>11</sup>

```
\g@addto@macro\verbatim{\microtypesetup{activate=false}}
```

*Compatibility.* The package is supposed to work happily together with all other  $\text{\LaTeX}$  packages (except for `pdfcpot`). However, life isn’t perfect, so problems are to be expected. Currently, I am only aware of the following issue:

- 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). If your document includes multiple input encodings, 8-bit characters in the settings will not work reliably unless you specify the `inputenc` key.

*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 `pdf $\text{\TeX}$`  in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your  $\text{\TeX}$  system.

<sup>11</sup> 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.

- ! pdfTeX error (font expansion): auto expansion is only possible with scalable fonts.  
Automatic font expansion has been improved in pdfTeX 1.40, in that it now not only works with Type 1 fonts but also with TrueType, OpenType and even non-embedded fonts. The above error message indicates either that you are trying to apply expansion to a bitmap (pk) font, which is still not possible, or that the font isn't found at all, e. g., because of missing map entries.
- Warning: pdflatex: font ptmr8r cannot be expanded (not an included Type1 font)  
and the PDF viewer complains about a missing font, e. g., Adobe Reader thusly:  
Could not find a font in the Resources dictionary - using Helvetica instead.  
With pdfTeX versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your TeX system is not set up to embed (or 'download') the base PostScript fonts (e. g., Times, Helvetica, Courier). In most TeX distributions, this can be changed in the file `upmap.cfg` by setting `pdftexDownloadBase14` to true.
- Warning: pdflatex (file ecrm1000+20): Font ecrm1000+20 at 1200 not found  
Furthermore, pdfTeX versions older than 1.40 require Type 1 fonts for automatic font expansion. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. With older pdfTeX versions, this is only possible if you manually create expanded instances of the fonts.
- ! Font Tl/cmr/m/n/10=ecrm1000 at 10.0pt not loaded: Not enough room left.  
Memory parameter 'font\_mem\_size' too small.
- ! TeX capacity exceeded, sorry [maximum internal font number (font\_max)=2000].  
Memory parameter 'font\_max' too small.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf\_mem\_size)=65536].  
Memory parameter 'pdf\_mem\_size' too small (pdfTeX versions older than 1.30).  
When applying micro-typographic enhancement to a large document with a lot of fonts, pdfTeX may be running out of some kind of memory. It can be increased by setting the respective parameter to a larger value. For web2c-based systems, change the settings in `texmf.cnf`, for MiKTeX, in the file `miktex.ini` (2.4 or older) resp. `pdflatex.ini` (2.5 or newer).
- pdfTeX warning (font expansion): font should be expanded before its first use  
This warning will occur with pdfTeX versions older than 1.40.4, if tracking *and* expansion is applied to a font. It is harmless and can be ignored.

## 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 $\TeX$  programme in the first place, which introduced the micro-typographic extensions and made them available to the  $\TeX$  world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004]. I also thank him and the rest of the pdf $\TeX$  team for refuting the idea that  $\TeX$  is dead, and for fixing the bugs I find.

*Harald Harders* has contributed protrusion settings for Adobe Minion. I would also like to thank him for a number of bug reports and suggestions he had to make. *Andreas Böhmann* has suggested the possibility to specify ranges of font sizes, and resourcefully assisted in implementing this. He also came up with some good ideas for the management of complex configurations. *Ulrich Dirr* has made numerous suggestion, especially concerning the new extensions of interword spacing adjustment and additional character kerning. 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* and *Peter Meier*.

## 12 References

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

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

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, Martin Schröder, *The pdf $\TeX$  user manual*, 25 January 2007. (Available from CTAN at [/systems/pdftex/](http://systems/pdftex/); latest version at <http://sarovar.org/projects/pdftex/>)

Karl Berry, *Fontname. Filenames for  $\TeX$  fonts*, September 2005. (Available from CTAN at [/info/fontname/fontname.pdf](http://info/fontname/fontname.pdf))

L<sup>A</sup> $\TeX$ 3 Project Team, *L<sup>A</sup> $\TeX$ 2 $\epsilon$  font selection*, 27 November 2005. (Available from CTAN at [/macros/latex/doc/fntguide.pdf](http://macros/latex/doc/fntguide.pdf))

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

Melchior Franz, *The soul package*, 17 November 2003. (Available from CTAN at [/macros/latex/contrib/soul/](#))

## 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.

### 2.3 (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
- Basic support for Lua $\TeX$  (if generated with the `lua` option)

### 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 `\lstyle` 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 1]

- 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)
- ‘DVloutput’ option will work with T<sub>E</sub>XLive 2004

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
- 8-bit characters in the configuration finally work as advertised, even if made active by the csquotes package
- 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
- Disable expansion inside `\showhyphens`
- 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]
- Protrusion settings of digits improved
- 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

## 1.4b (2004/11/26)

- `\UseMicrotypeSet` requires the set to be declared [section 4]

## 1.4 (2004/11/12)

- Set up fonts independently from  $\LaTeX$  font loading (therefore, no risk of overlooking fonts anymore, and the package may be loaded at any time)
- New option: ‘`final`’ [section 3.5]

## 1.2 (2004/10/03)

- New font sets: ‘`allmath`’ and ‘`basicmath`’ [section 4; table 1]
- 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 Lua<sub>T</sub><sub>E</sub>X (by default disabled).

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}

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

5 <*package>
6 \newcommand*\DeclareMicrotypeSet[3] [] {}
7 \newcommand*\UseMicrotypeSet[2] [] {}
8 \newcommand*\DeclareMicrotypeSetDefault[2] [] {}
9 \newcommand*\SetProtrusion[3] [] {}
10 \newcommand*\SetExpansion[3] [] {}
11 \newcommand*\SetTracking[3] [] {}
12 \newcommand*\SetExtraKerning[3] [] {}
13 \newcommand*\SetExtraSpacing[3] [] {}
14 \newcommand*\DisableLigatures[2] [] {}
15 \newcommand*\DeclareCharacterInheritance[3] [] {}
16 \newcommand*\DeclareMicrotypeVariants[1] {}
17 \newcommand*\DeclareMicrotypeAlias[2] {}
18 \newcommand*\LoadMicrotypeFile[1] {}
19 \newcommand*\DeclareMicrotypeBabelHook[2] {}
20 \newcommand*\microtypesetup[1] {}
21 \newcommand*\microtypecontext[1] {}
22 \newcommand*\textmicrotypecontext[2] {#2}
23 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
24 </package>
25 \newcommand*\lststyle{}
26 \newcommand*\textls[2] [] {}
27 \def\textls#1#{
28 \newcommand*\lstlig[1] {#1}
29 <*package>
30 }
```

These commands also have a starred version.

```

31 \def\DeclareMicrotypeSet#1#\@gobbletwo{
32 \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.

```

33 \@onlypreamble\DeclareMicrotypeSet
34 \@onlypreamble\UseMicrotypeSet
35 \@onlypreamble\DeclareMicrotypeSetDefault
36 \@onlypreamble\DisableLigatures
37 \@onlypreamble\DeclareMicrotypeVariants
38 \@onlypreamble\DeclareMicrotypeBabelHook
```

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

```

39 \def\MT@old@cmd#1#2{%
40   \newcommand*#1{\MT@warning{%
41     \string#1 is deprecated. Please use\MessageBreak
42     \string#2 instead}%
43     \let #1#2#2}}

44 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
45 \MT@old@cmd\DeclareMicroTypeSet \DeclareMicrotypeSet
46 \MT@old@cmd\UseMicroTypeSet \UseMicrotypeSet
```

```

47 \MT@oldcmd\LoadMicroTypeFile \LoadMicrotypeFile
48 </package>

\MT@warning Communicate.
\MT@warning@nl 49 \def\MT@warning{\PackageWarning\MT@MT}
\MT@info 50 \def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
51 <+package>
\MT@info@nl 52 \def\MT@info{\PackageInfo\MT@MT}
\MT@vinfo 53 \def\MT@info@nl#1{\MT@info{#1\@gobble}}
\MT@error 54 \let\MT@vinfo\@gobble
55 \def\MT@error{\PackageError\MT@MT}
\MT@warn@err 56 \def\MT@warn@err#1{\MT@error{#1}{%
57 This error message appears because you loaded the \MT@MT\MessageBreak
58 package with the option \verbose=errors'. Consult the documentation\MessageBreak
59 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
```

```

60 <+debug>
61 \MT@warning@nl{This is the debug version}
62 \newcount\tracingmicrotype
63 \tracingmicrotype=2
64 \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
65 \def\MT@info@nl#1{\PackageInfo\MT@MT{#1\@gobble}\MT@addto@annot{#1}}
66 \let\MT@vinfo\MT@info@nl
67 \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
68 \def\MT@warning@nl#1{\PackageWarning\MT@MT{#1\@gobble}\MT@addto@annot{Warning: #1}}
69 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
70 \def\MT@dinfo@nl#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@nl{#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
```

```
71 \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 pdfTeX ≥ 1.30.)

```

\MT@addto@annot
\ifMT@inannot 72 \newif\ifMT@inannot \MT@inannottrue
73 \let\MT@pdf@annot\@empty
74 \def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
75 {\def\MessageBreak{^^J\@spaces}%
76 \MT@xadd\MT@pdf@annot{\pdfescapestring{#1^^J}}}\fi\fi}
```

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

77 `\newif\iftracingmicrotypeinpdfall`

`\MT@showpdfannot` 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).

```

78 \def\MT@showpdfannot#1{%
79   \ifnum\tracingmicrotypeinpdf<#1 \else
80     \iftracingmicrotypeinpdfall\leavevmode\fi
81     \pdfannot height 4pt width 4pt depth 2pt {%
82       /Subtype/Caret
83       /T(\expandafter\string\font@name)
84       \ifcase#1\or
85         /Subj(New font)/C[1 0 0]
86       \else
87         /Subj(Known font)/C[0 1 0]
88       \fi
89       /Contents(\MT@pdf@annot)
90     }%
91     \iftracingmicrotypeinpdfall\kern1pt \fi
92     \global\MT@inannotfalse
93   \fi
94 }
95 </debug>
96 </package>

```

### 14.1.2 Requirements

`\MT@plain` The `letterspace` package also works with `eplain` (1) or `miniltx` (2).

```

97 <*plain>
98 \def\MT@plain{0}
99 \ifx\documentclass\@undefined
100   \def\MT@plain{1}
101   \def\hmode\bgroup{\leavevmode\bgroup}
102   \let\@typeset@protect\relax
103   \ifx\epain\@undefined
104     \def\MT@plain{2}
105     \def\PackageWarning#1#2{%
106       \begingroup
107         \newlinechar=10 %
108         \def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces\@spaces}%
109         \immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}%
110       \endgroup
111     }
112     \def\on@line{ on input line \the\inputlineno}
113     \def\@spaces{\space\space\space\space}
114   \fi
115 \fi
116 </plain>

```

`\MT@pdf@tex@no` pdf<sub>T</sub><sub>E</sub><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><sub>E</sub><sub>X</sub> we're using, if any. `\MT@pdf@tex@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><sub>E</sub><sub>X</sub>:

- 0: not running pdf<sub>TeX</sub>
- 1: pdf<sub>TeX</sub> (< 0.14f)
- 2: + micro-typographic extensions (0.14f,g)
- 3: + protrusion relative to 1em ( $\geq$  0.14h)
- 4: + automatic font expansion; protrusion no longer has to be set up first; default `\efcode = 1000` ( $\geq$  1.20)
- 5: + `\(left,right)marginkern`; `\pdfnoligatures`; `\pdfstrcmp`; `\pdfescapestring` ( $\geq$  1.30)
- 6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`<sup>12</sup>; `\pdftracingfonts`; always e-<sub>TeX</sub> ( $\geq$  1.40)
- 7: + `\letterspacefont` doesn't disable ligatures and kerns; `\pdfcopyfont` ( $\geq$  1.40.4)

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

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

```
118 \ifx\normalpdftexversion\@undefined \else
119   \let\pdftexversion\normalpdftexversion
120   \let\pdftexrevision\normalpdftexrevision
121   \let\pdfoutput\normalpdfoutput
122 \fi
```

Old packages might have let `\pdftexversion` to `\relax`.

```
123 \ifx\pdftexversion\@undefined \else
124   \ifx\pdftexversion\relax \else
125     <debug>\MT@info@nl{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
126     \def\MT@pdftex@no{7}
127   <*package>
128     \ifnum\pdftexversion = 140
129       \ifnum\pdftexrevision < 4
130         \def\MT@pdftex@no{6}
131       \fi
132     \else
133   </package>
134     \ifnum\pdftexversion < 140
135       \def\MT@pdftex@no{5}
136   <*package>
137     \ifnum\pdftexversion < 130
138       \def\MT@pdftex@no{4}
139     \ifnum\pdftexversion < 120
140       \def\MT@pdftex@no{3}
141     \ifnum\pdftexversion = 14
142       \ifnum\expandafter`\pdftexrevision < `h
143         \def\MT@pdftex@no{2}
144       \ifnum\expandafter`\pdftexrevision < `f
145         \def\MT@pdftex@no{1}
146       \fi
147     \fi
148   \else
149     \ifnum\pdftexversion < 14
150       \def\MT@pdftex@no{1}
151     \fi
```

<sup>12</sup> This command was actually introduced in 1.30, but failed on strings longer than 1023 bytes.

```

152         \fi
153     \fi
154     \fi
155     \fi
156 </package>
157     \fi
158     \fi
159 \fi
160 <debug>\MT@info@nl{0}{pdfTeX no.: \MT@pdfTeX@no}

```

\MT@clear@options     If we are not using pdfTeX or in case it is too old, we disable everything and exit here.

```

161 \def\MT@clear@options{%
162 <plain> \ifnum\MT@plain<2
163   \AtEndOfPackage{\let\@unprocessedoptions\relax}%
164   \let\CurrentOption\@empty
165 <plain> \fi
166 }
167 \ifnum\MT@pdfTeX@no <
168 <package> 2
169 <letterspace> 6
170   \MT@clear@options
171   \MT@warning@nl{You
172     \ifcase\MT@pdfTeX@no
173       don't seem to be using pdfTeX.\MessageBreak
174       ~\MT@MT' only works with pdfTeX.\MessageBreak
175       Try running ~pdfLaTeX' instead of ~LaTeX'%
176     \else
177       are using a pdfTeX version older than
178 <package> 0.14f%
179 <letterspace> 1.40%
180       .\MessageBreak
181       ~\MT@MT' does not work with this version.\MessageBreak
182       Please install a newer version of pdfTeX%
183     \fi
184   }
185 \endinput\fi
186 <*package>

```

Still there? Then we can begin:

\MT@catcodes     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.

```

187 \def\MT@catcodes{%
188   \catcode~\^7 %
189   \@makeother\~%
190   \@makeother\=%
191   \@makeother\*%
192   \@makeother\+%
193   \@makeother\,%
194   \@makeother\/%
195   \@makeother\`%
196   \@makeother\'%
197   \@makeother\"%
198   \@makeother\!%
199 }

```

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

```

200 \def\MT@restore@catcodes#1{%
201   \ifx\relax#1\else
202     \noexpand\catcode~\noexpand#1\the\catcode~#1\relax

```

```

203 \expandafter\MT@restore@catcodes
204 \fi
205 }
206 \edef\MT@restore@catcodes{\MT@restore@catcodes\^-\-=\*+\.\/\`\'"\!\\relax}
207 \MT@catcodes
208 \AtEndOfPackage{\MT@restore@catcodes}

```

We need the `keyval` package, including the ‘new’ `\KV@sp@def` implementation.

```

209 </package>
210 \RequirePackage{keyval}[1997/11/10]
211 <*package>

```

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

```
212 \newtoks\MT@toks
```

`\ifMT@if@` A scratch if.

```
213 \newif\ifMT@if@
```

### 14.1.3 Declarations

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

```

\ifMT@expansion 214 \newif\ifMT@protrusion
\ifMT@auto 215 \newif\ifMT@expansion
\ifMT@selected 216 \newif\ifMT@auto
\ifMT@selected 217 \newif\ifMT@selected
\ifMT@noligatures 218 \newif\ifMT@noligatures
\ifMT@draft 219 \newif\ifMT@draft
\ifMT@spacing 220 \newif\ifMT@spacing
\ifMT@kerning 221 \newif\ifMT@kerning
\ifMT@tracking 222 \newif\ifMT@tracking
\ifMT@tracking 223 \newif\ifMT@babel

```

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

```

\MT@ex@level 224 \let\MT@pr@level\tw@
\MT@pr@factor 225 \let\MT@ex@level\tw@
\MT@ex@factor 226 \let\MT@pr@factor@m
\MT@ex@factor 227 \let\MT@ex@factor@m
\MT@sp@factor 228 \let\MT@sp@factor@m
\MT@kn@factor 229 \let\MT@kn@factor@m

```

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

```

\MT@kn@unit 230 \let\MT@pr@unit\@empty
\MT@stretch 231 \let\MT@sp@unit\m@ne
\MT@shrink 232 \def\MT@kn@unit{1em}

```

`\MT@step` Expansion settings.

```

233 \let\MT@stretch\m@ne
234 \let\MT@shrink \m@ne
235 \let\MT@step \m@ne

```

`\MT@pr@min` Minimum and maximum values allowed by pdf<sub>TeX</sub>.

```

\MT@pr@max 236 \def\MT@pr@min{-\@m}
\MT@ex@min 237 \let\MT@pr@max\@m
238 \let\MT@ex@min\z@
\MT@ex@max 239 \let\MT@ex@max\@m
\MT@sp@min 240 \def\MT@sp@min{-\@m}
\MT@sp@max 241 \let\MT@sp@max\@m
242 \def\MT@kn@min{-\@m}
\MT@kn@min 243 \let\MT@kn@max\@m
\MT@kn@max
\MT@tr@min
\MT@tr@max

```

```

244 </package>
245 \def\MT@tr@min{-\@m}
246 \let\MT@tr@max\@m
247 <*package>

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

\MT@stretch@default     Default values for expansion.
\MT@shrink@default      249 \def\MT@stretch@default{20 }
\MT@step@default        250 \def\MT@shrink@default{20 }
251 \def\MT@step@default{4 }

\MT@letterspace         Default value for letterspacing (in thousandths of 1em).
\MT@letterspace@default 252 </package>
253 \let\MT@letterspace\m@ne
254 \def\MT@letterspace@default{100}
255 <*package>

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

```

#### 14.1.4 Auxiliary Macros

```

\MT@maybe@etex        For definitions that depend on e-TeX features.
\MT@maybe@etex@       257 \def\MT@maybe@etex{%
258   \ifcase 0%
259     \ifx\TeXversion\undefined 1\else
260       \ifx\TeXversion\relax 1\else
261         \ifcase\TeXversion 1\fi
262       \fi
263     \fi
264     \catcode`Q=14 \catcode`X=9
265   \else
266     \catcode`Q=9 \catcode`X=14
267   \fi
268   \MT@maybe@etex@
269 }
270 \def\MT@maybe@etex@#1{#1%
271   \catcode`Q=11 \catcode`X=11
272 }
273 <debug>\MT@maybe@etex{\MT@info@n1{0}}{this is
274 <debug>Q not
275 <debug> etex}}

\MT@requires@pdftex    For definitions that depend on a particular pdfTeX version.
276 \def\MT@requires@pdftex#1{%
277   \ifnum\MT@pdftex@no<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
278 }
279 <debug>\MT@requires@pdftex6{\pdftracingfonts=1}\relax

\MT@requires@luatex    For definitions that depend on LuaTeX.
280 <*lua>
281 \def\MT@requires@luatex{%
282   \ifx\directlua\undefined\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
283 }
284 <debug>\MT@info@n10{this is \MT@requires@luatex}{\not }luatex}

\MT@lua                Communicate with lua.
285 \def\MT@lua{\directlua0}

```

```
286 </lua>
287 </package>
```

Since LuaTeX is still heavily experimental, we don't support it by default. Also, at the time of this writing, LuaTeX ignores `\letterspacefont`.

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

`\MT@glet` The forgotten primitive.

```
301 \def\MT@glet{\global\let}
302 <!*package>
```

`\MT@def@n` This is `\@namedef` and global.

```
\MT@gdef@n 303 \def\MT@def@n#1{\expandafter\def\csname #1\endcsname}
304 \def\MT@gdef@n#1{\expandafter\gdef\csname #1\endcsname}
```

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

```
\MT@xdef@n 305 </package>
306 \def\MT@edef@n#1{\expandafter\edef\csname #1\endcsname}
307 <!*package>
308 \def\MT@xdef@n#1{\expandafter\xdef\csname #1\endcsname}
```

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

```
309 \def\MT@let@nc#1{\expandafter\let\csname #1\endcsname}
```

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

```
310 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}
```

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

```
311 \def\MT@let@nn#1{\expandafter\MT@let@cn\csname #1\endcsname}
```

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

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

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

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

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

```
315 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
316 <!*package>
```

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

```
317 \def\MT@exp@two@n#1#2#3{%
318 \expandafter\expandafter\expandafter
319 #1\expandafter\expandafter\expandafter
320 {\expandafter#2\expandafter}\expandafter{#3}}
```

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



<code>\MT@ifdefined@c@T</code>	Wrapper for testing whether command resp. <code>\csname</code> sequence is defined. If we
<code>\MT@ifdefined@c@TF</code>	are running e- $\TeX$ , we will use its primitives <code>\ifdefined</code> and <code>\ifcsname</code> , which
<code>\MT@ifdefined@n@T</code>	decreases memory use substantially.
<code>\MT@ifdefined@n@TF</code>	<pre> 321 \MT@maybe@etex{ 322   \def\MT@ifdefined@c@T#1{% 323     \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi 324     \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi 325   } 326   \&lt;/package&gt; 327   \&lt;letterspace&gt;\catcode`\X=9 328   \def\MT@ifdefined@c@TF#1{% 329     \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi 330     \&lt;package&gt;Q \ifx#1\@undefined 331     \&lt;package&gt;Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi 332   } 333   \def\MT@ifdefined@n@T#1{% 334     \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi 335     \&lt;package&gt;Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax 336     \&lt;package&gt;Q \expandafter\@gobble\else\expandafter\@firstofone\fi 337   } 338   \&lt;letterspace&gt;\catcode`\X=11 339   \&lt;*package&gt; 340   \def\MT@ifdefined@n@TF#1{% 341     \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi 342     Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax 343     Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi 344   } 345 } </pre>
<code>\MT@detokenize@n</code>	Translate a macro into a token list. With e- $\TeX$ , we can use <code>\detokenize</code> (and
<code>\MT@detokenize@c</code>	<code>\expandafter\string</code> to get rid of the trailing space). The $\epsilon$ - $\TeX$ version requires
	some more fiddling (and the <code>\string</code> isn't perfect, of course).
	<pre> 346 \MT@maybe@etex{ 347   \def\MT@detokenize@n#1{% 348     X \detokenize\expandafter{% 349       \string#1% 350     }% 351   } 352   \def\MT@detokenize@c#1{% 353     X \detokenize\expandafter\expandafter\expandafter\string#1}% 354     \MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty 355   } 356 } </pre>
<code>\MT@ifempty</code>	Test whether argument is empty.
	<pre> 357 \&lt;/package&gt; 358 \begingroup 359 \catcode`\%=12 360 \catcode`\&amp;=14 361 \gdef\MT@ifempty#1{% 362   \if %#1%&amp; 363     \expandafter\@firstoftwo 364   \else 365     \expandafter\@secondoftwo 366   \fi 367 } 368 \endgroup 369 \&lt;*package&gt; </pre>
<code>\MT@ifint</code>	Test whether argument is an integer, using an old trick by Mr. Arseneau, or the
	latest and greatest from pdf $\TeX$ or Lua $\TeX$ (which also allows negative numbers, as

required by the letterspace option).

```

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

`\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.)

```

402 \MT@requires@pdftex6{
403   \lua
404   \MT@requires@luatex{
405     \def\MT@ifdimen#1{%
406       \csname
407         \MT@lua{if (string.find("\luaescapestring{#1}", "^-[0-9]+(\@percentchar a*) *$") or
408           string.find("\luaescapestring{#1}", "^-[0-9]*[.,][0-9]+(\@percentchar a*) *$"))
409           then tex.write("@firstoftwo")
410           else tex.write("@secondoftwo")
411         end}%
412     \endcsname
413   }
414 }{
415 \lua
416 \def\MT@ifdimen#1{%
417   \ifcase\pdfmatch{^([0-9]+([.,][0-9]+)?|[.,][0-9]+)%
418     (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
419   \expandafter\@secondoftwo
420   \else
421     \expandafter\@firstoftwo
422   \fi
423 }
424 \lua }
425 }{
426   \def\MT@ifdimen#1{%
427     \setbox\z@=\hbox{#1}
```

```

428     \MT@count=1#1\relax
429     \ifnum\MT@count=\@ne
430         \aftergroup\@secondoftwo
431     \else
432         \aftergroup\@firstoftwo
433     \fi
434 }%
435 }
436 }

\MT@ifdim    Test floating point numbers.
437 \def\MT@ifdim#1#2#3{%
438     \ifdim #1\p@ #2 #3\p@
439         \expandafter\@firstoftwo
440     \else
441         \expandafter\@secondoftwo
442     \fi
443 }

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

\MT@xadd    Add item to a list.
476 \def\MT@xadd#1#2{%
477     \ifx#1\relax
478         \xdef#1{#2}%
479     \else
480         \xdef#1{#1#2}%
481     \fi
482 }

```

```

\MT@xaddb      Add item to the beginning.
483 \def\MT@xaddb#1#2{%
484   \ifx#1\relax
485     \xdef#1{#2}%
486   \else
487     \xdef#1{#2#1}%
488   \fi
489 }

\MT@map@clist@n  Run <#2> on all elements of the comma list <#1>. This and the following is modelled
\MT@map@clist@c  after LATEX3 commands.
\MT@map@clist@ 490 </package>
\MT@clist@function 491 \def\MT@map@clist@n#1#2{%
492   \ifx\@empty#1\else
\MT@clist@break 493     \def\MT@clist@function##1{#2}%
494     \MT@map@clist@#1,\@nil,\@nnil
495   \fi
496 }
497 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
498 \def\MT@map@clist@#1,{%
499   \ifx\@nil#1%
500     \expandafter\MT@clist@break
501   \fi
502   \MT@clist@function{#1}%
503   \MT@map@clist@
504 }
505 \let\MT@clist@function\@gobble
506 \def\MT@clist@break#1\@nnil{}
507 <*package>

\MT@map@tlist@n  Execute <#2> on all elements of the token list <#1>. \MT@tlist@break can be used
\MT@map@tlist@c  to jump out of the loop.
\MT@map@tlist@ 508 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
509 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
\MT@tlist@break 510 \def\MT@map@tlist@#1#2{%
511   \ifx\@nnil#2\else
512     #1{#2}%
513     \expandafter\MT@map@tlist@
514     \expandafter#1%
515   \fi
516 }
517 \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 518 \newif\ifMT@inlist@
519 \def\MT@in@clist#1#2{%
520   \def\MT@res@a#1,#1,##2##3\@nnil{%
521     \ifx##2\@empty
522       \MT@inlist@false
523     \else
524       \MT@inlist@true
525     \fi
526   }%
527   \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
528 }

\MT@rem@from@clist  Remove item <#1> from comma list <#2>. This is basically \@removeelement from
ltcntrl.dtx. Using \pdfmatch and \pdfmatch here would be really slow!
529 \def\MT@rem@from@clist#1#2{%
530   \def\MT@res@a#1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
531   \def\MT@res@b#1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%

```

```

532 \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
533 }

\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.

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

\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@@  546 \def\MT@in@rlist#1{%
\MT@size@name    547   \MT@inlist@false
548   \MT@map@tlist@c#1\MT@in@rlist@
549 }
550 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
551 \def\MT@in@rlist@@#1#2#3{%
552   \MT@ifdim{#2}=\m@ne{%
553     \MT@ifdim{#1}=\MT@size
554     \MT@inlist@true
555     \relax
556   }{%
557     \MT@ifdim\MT@size<{#1}\relax{%
558       \MT@ifdim\MT@size<{#2}%
559       \MT@inlist@true
560       \relax
561     }%
562   }%
563   \ifMT@inlist@
564     \def\MT@size@name{#3}%
565     \expandafter\MT@tlist@break
566   \fi
567 }

\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      568 </package>
569 \def\MT@loop#1\MT@repeat{%
570   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
571   \MT@iterate \let\MT@iterate\relax
572 }
573 \let\MT@repeat\fi

\MT@while@num   Execute <#3> from <#1> up to (excluding) <#2>.
574 \def\MT@while@num#1#2#3{%
575   \@tempcnta#1\relax
576   \MT@loop #3%
577   \advance\@tempcnta \@ne
578   \ifnum\@tempcnta < #2\MT@repeat
579 }

```

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

```
580 \def\MT@do@font{\MT@while@num\z@\cc\vi}
581  $\langle *package \rangle$ 
```

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

```
\MT@increment
582 \newcount\MT@count
583 \MT@maybe@etex{
584   \def\MT@increment#1{%
585     \edef#1{\number\numexpr #1 + 1\relax}%
586   }
587   \advance\MT@count \@ne
588   \edef#1{\number\MT@count}%
589 }
```

`\MT@scale`      Multiply and divide a counter. If we are using e-TeX, we will use its `\numexpr` primitive. This has the advantage that it is less likely to run into arithmetic overflow. The result of the division will be rounded instead of truncated. Therefore, we'll get a different (more accurate) result in about half of the cases.

```
590 \def\MT@scale#1#2#3{%
591   \multiply #1 #2\relax
592   \ifnum #3 = \z@
593     #1=\numexpr #1 * #2\relax
594   \else
595     #1=\numexpr #1 * #2 / #3\relax
596   \divide #1 #3\relax
597   \fi
598 }
599 }
```

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

```
\MT@abbr@ex
\MT@abbr@pr@c 600 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 601 \def\MT@abbr@ex{expansion}
\MT@abbr@pr@inh 602 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@ex@inh 603 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh 604 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@n 605 \def\MT@abbr@ex@inh{expansion inheritance}
\MT@abbr@n 606 \def\MT@abbr@n{noligatures}
\MT@abbr@sp 607 \def\MT@abbr@sp{spacing}
\MT@abbr@sp@c 608 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@sp@inh 609 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@kn 610 \def\MT@abbr@kn{kerling}
\MT@abbr@kn@c 611 \def\MT@abbr@kn@c{kerling codes}
\MT@abbr@kn@c 612 \def\MT@abbr@kn@inh{kerling inheritance}
\MT@abbr@kn@c 613 \def\MT@abbr@tr{tracking}
\MT@abbr@kn@c 614 \def\MT@abbr@tr@c{tracking amount}
```

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

```
\MT@rbba@protrusion
\MT@rbba@tr@c
\MT@rbba@expansion 615 \def\MT@rbba@protrusion{pr}
\MT@rbba@spacing 616 \def\MT@rbba@expansion{ex}
\MT@rbba@kerling 617 \def\MT@rbba@spacing{sp}
\MT@rbba@tracking 618 \def\MT@rbba@kerling{kn}
\MT@rbba@tracking 619 \def\MT@rbba@tracking{tr}
```

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

```
\MT@features@long 620 \def\MT@features{pr,ex,sp,kn,tr}
621 \def\MT@features@long{protrusion,expansion,spacing,kerling,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.

```

622 \def\MT@is@feature#1{%
623   \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
624   \ifMT@inlist@
625     \expandafter\@firstofone
626   \else
627     \MT@error{\@tempa' is not an available micro-typographic\MessageBreak
628       feature. Ignoring #1}{Available features are: \MT@features@long'.}%
629     \expandafter\@gobble
630   \fi
631 }
```

### 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`
- `\showhyphens`

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.

```

632 \ifl@aded{tex}{wordcount}{%
633   \MT@warning@nl{Detected the `wordcount' utility.\MessageBreak
634     Disabling `~\MT@MT', since it wouldn't work}%
635   \MT@clear@options\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.

```

636 </package>
637 <plain>\ifnum\MT@plain<2
638 \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.

```

639 \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).

```

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

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

```

641 \def\MT@with@package@T#1{\ifpackage@loaded{#1}\@firstofone\@gobble}
642 <*package>
```

`\MT@with@babel@and@T` L<sup>A</sup>T<sub>E</sub>X’s `\ifpackage@with` 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 off the lines one by one. This will destroy character protrusion. (There aren't any problems with the lineno package, since it takes a different approach.) — ... — After much to and fro, the situation has finally settled and there is a fix. Beginning with pdfTeX version 1.21b together with ledpatch.sty as of 2005/06/02 (v0.4), character protrusion will work at last.

Peter Wilson was so kind to provide the \l@dunhbox@line hook in ledmac to allow for protrusion. \leftmarginkern and \rightmarginkern are new primitives of pdfTeX 1.21b (aka. 1.30.0).

```

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 babel modify `\%`, storing the original meaning in `\percentsign`.

```
686 \MT@with@babel@and@T{spanish} {\MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
687 \MT@with@babel@and@T{galician} {\MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
```

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.)

```
688 \MT@with@package@T{csquotes}
689 {\@ifpackagelater{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`.

```
690 \@ifpackageloaded{hyperref} \MT@restore@p@h
691 {\MT@with@package@T{tex4ht} \MT@restore@p@h}%
692 }
```

Check again at the end of the preamble.

```
693 </package>
694 \MT@addto@setup{%
695 <*package>
```

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

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

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

```
704 \let\MT@setupfont@hook\empty
705 \MT@with@babel@and@T{spanish}{%
706 \g@addto@macro\MT@setupfont@hook{%
707 \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
708 }
709 \MT@with@babel@and@T{galician}{%
710 \g@addto@macro\MT@setupfont@hook{%
711 \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
712 }
713 \MT@with@package@T{csquotes}{%
714 \@ifpackagelater{csquotes}{2005/05/11}{%
715 \g@addto@macro\MT@setupfont@hook\@disablequotes
716 }{%
717 \MT@warning@n1{%
718 Should you receive warnings about unknown slot\MessageBreak
719 numbers, try upgrading the 'csquotes' package}%
720 }%
721 }
722 \@ifpackageloaded
723 </package>
724 <letterspace> \MT@with@package@T
725 {hyperref}{%
726 <package> \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
```

We disable microtype's additions inside hyperref's `\pdfstringdef`, which redefines lots of commands.

```

727 \pdfstringdefDisableCommands{%
728 <*package>
729 \let\pickup@font\MT@orig@pickupfont
730 \let\textmicrotypecontext\@secondoftwo
731 \let\microtypecontext\@gobble
732 </package>
733 \def\lsstyle{\pdfstringdefWarn\lsstyle}%
734 \def\textls#1{\pdfstringdefWarn\textls}%
735 }%
736 }
737 <*package>
738 {\MT@with@package@T{tex4ht}
739 {\g@addto@macro\MT@setupfont@hook\MT@restore@p@h}}%
```

The `listings` package makes numbers and letters active,

```

740 \MT@with@package@T{listings}{%
741 \g@addto@macro\MT@cfg@catcodes{%
742 \MT@while@num{"30}{\catcode\@tempcnta 12\relax}%
743 \MT@while@num{"41}{\catcode\@tempcnta 11\relax}%
744 \MT@while@num{"61}{\catcode\@tempcnta 11\relax}%
745 }%
```

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

```

746 \g@addto@macro\MT@setupfont@hook{%
747 \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.

```

748 \let\lst@ProcessLetter\empty
749 }%
750 }
```

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.

```

751 %
752 </package>
753 \MT@with@package@T{soul}{%
754 \soulregister\lsstyle 0%
755 \soulregister\textls 1%
756 }
757 <*package>
```

Compatibility with `pinyin` (CJK) package: disable `microtype` in `\py@macron`, which loads a different font for the accent. In older versions of `pinyin`, `\py@macron` had only one argument.

```

758 \MT@with@package@T{pinyin}{%
759 \let\MT@orig@py@macron\py@macron
760 \ifpackageafter{pinyin}{2006/10/17}{% 4.7.0
761 \def\py@macron#1#2{%
762 \let\pickup@font\MT@orig@pickupfont
763 \MT@orig@py@macron{#1}{#2}%
764 \let\pickup@font\MT@pickupfont}%
765 }{%
766 \def\py@macron#1{%
767 \let\pickup@font\MT@orig@pickupfont
768 \MT@orig@py@macron{#1}%
```

```

769     \let\pickup@font\MT@pickupfont}%
770   }%
771 }
772 </package>
773 }

```

The miniltx version.

```

774 <*plain>
775 \else
776   \let\MT@addto@setup\@firstofone
777 \fi
778 </plain>
779 <*package>

```

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

```

780 \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`).

```

781 \def\MT@setupfont{%

```

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

```

782   \MT@setupfont@hook
783 }

```

This will use a copy of the font (allowing for expansion parameter variation).

```

784 \MT@requires@pdftex7
785 {\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!

```

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

```

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

```

788   \MT@exp@one@n\MT@find@file\MT@family
789   \ifx\MT@familyalias\@empty \else
790     \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.)

```

791 % \ifx\f@encoding\cf@encoding\else\@enc@update\fi
792 }

```

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

```

793 \MT@requires@pdftex6
794 {\g@addto@macro\MT@setupfont\MT@tracking}\relax
795 \g@addto@macro\MT@setupfont{%
796   \MT@check@font
797   \ifMT@inlist@
798     <debug>\MT@show@pdfannot2%
799   \else
800     \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.

```
801 \MT@protrusion
802 \MT@expansion
803 }
```

Interword spacing and kerning (pdfTeX 1.40).

```
804 \MT@requires@pdftex6
805 {\g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}}\relax
```

Disable ligatures (pdfTeX 1.30).

```
806 \MT@requires@pdftex5
807 {\g@addto@macro\MT@setupfont\MT@noligatures}\relax
808 \g@addto@macro\MT@setupfont{%
```

Debugging.

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

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

```
810 \MT@register@font
811 \fi
812 }
```

`\MT@copy@font` The new `\pdfcopyfont` command allows to expand a font with different parameters.  
`\MT@copy@font@` It will be used when we find a context for `\SetExpansion` in the preamble, or when the package has been loaded with the `copyfonts` option.

```
813 \MT@requires@pdftex7{
814 \let\MT@copy@font\relax
815 \def\MT@copy@font{%
```

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

```
816 \xdef\MT@font@copy{\csname\MT@font/\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.

```
817 \expandafter\ifx\MT@font@copy\relax
818 \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
819 \expandafter\ifx\MT@font@orig\relax
820 \MT@exp@two@c\MT@gl@et\MT@font@orig\font@name
821 \else
822 \MT@exp@two@c\let\font@name\MT@font@orig
823 \fi
824 \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
825 <debug>\MT@dinfol{creating new copy: \MT@font@copy}%
```

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

```
826 \MT@map@clist@c\MT@active@features{%
827 \expandafter\ifx\csname \MT@font@name\endcsname\relax\else
828 \def\@tempa{##1}%
829 \expandafter\MT@map@tlist@c
830 \csname \MT@font@name\endcsname
831 \MT@rem@from@list
832 \fi
833 }%
834 \fi
835 \MT@exp@two@c\let\MT@font\MT@font@copy
```

We only need the font identifier for letterspacing.

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

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

```
837 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
838 }
```

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

```
839 \def\MT@rem@from@list#1{%
840   \expandafter\ifx\csname MT@\@tempa @#1font@list\endcsname\relax\else
841     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
842       \MT@font \csname MT@\@tempa @#1font@list\endcsname
843   \fi
844 }
845 }\relax
```

*Here's the promised dirty trick* for users of older pdf<sub>T</sub><sub>E</sub>X 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 10pt:

```
\SetExpansion
[ stretch = 30,
  shrink = 60,
  step = 5 ]
{ encoding = *,
  size = 10.001 }
{ }
\newcommand{\expandpar}[1]{\{%
  \fontsize{10.001}{\baselineskip}\selectfont #1\par}}
% ...
\expandpar{This paragraph contains an 'unnecessary' widow.}
```

Note that the `\expandpar` command can only be applied to complete paragraphs. If you are using Computer Modern Roman, you have to load the `fix-cm` package to be able to select fonts in arbitrary sizes. Finally, the reason I suggest to use a larger font, and not a smaller one, is to prevent a different design size being selected.

`\MT@split@name` Split up the font name (`<#6>` may be an expansion context and/or a letterspacing amount).

```
\MT@encoding
\MT@family 846 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
847   \def\MT@encoding{#1}%
\MT@series 848   \def\MT@family {#2}%
\MT@shape 849   \def\MT@series {#3}%
\MT@size 850   \def\MT@shape {#4}%
851   \def\MT@size {#5}%
```

`\MT@familyalias` Alias family?

```
852 \MT@ifdefined@n@TF{MT@\MT@family @alias}%
853 { \MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
854 { \let\MT@familyalias\@empty}%
855 }
```

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

`\MT@feat`

`\MT@maybe@do`

```

856 \newif\ifMT@do
857 \def\MT@maybe@do#1{%

```

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

```

858 \expandafter\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).

```

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

```

`\MT@feat` stores the current feature.

```

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

```

`\MT@checklist@` The generic test (`<#1>` is the axis, `<#2>` the feature, `\@tempa` contains the set name).

```

877 \def\MT@checklist@#1#2{%
878 <debug> \MT@ifdefined@n@T
879 <debug> \MT@ifdefined@n@TF
880 {MT@#2list@#1@\@tempa}{%

```

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

```

881 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
882 \csname MT@#1\expandafter\endcsname
883 \csname MT@#2list@#1@\@tempa\endcsname
884 \ifMT@inlist@
885 <debug>\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#2}: #1 ~\@nameuse{MT@#1}' in list}%
886 \MT@dotrue
887 \else
888 <debug>\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#2}: #1 ~\@nameuse{MT@#1}' not in list}%
889 \MT@dofalse
890 \expandafter\MT@clist@break
891 \fi
892 }%

```

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.

```

893 <debug> {\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#2}: #1 list empty}}%
894 }

```

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

```

895 \def\MT@checklist@family#1{%
896 <debug> \MT@ifdefined@n@T
897 <debug> \MT@ifdefined@n@TF
898 {MT@#1list@family@\@tempa}{%
899 \MT@exp@two@n\MT@in@clist

```

```

900     \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
901     \ifMT@inlist@
902 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family `~\@nameuse{MT@family}' in list}%
903     \MT@dotrue
904     \else
905 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family `~\@nameuse{MT@family}' not in list}%
906     \MT@dofalse
907     \ifx\MT@familyalias\@empty \else
908     \MT@exp@two@n\MT@in@clist
909     \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
910     \ifMT@inlist@
911 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: alias `~\MT@familyalias' in list}%
912     \MT@dotrue
913 <debug> \else\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: alias `~\MT@familyalias' not in list}%
914     \fi
915     \fi
916     \fi
917     \ifMT@do \else
918     \expandafter\MT@clist@break
919     \fi
920 }%
921 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family list empty}}%
922 }

```

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

```

923 \def\MT@checklist@size#1{%
924 <!debug> \MT@ifdefined@n@T
925 <debug> \MT@ifdefined@n@TF
926 {MT@#1list@size@\@tempa}%
927 \expandafter\MT@in@rlist\csname MT@#1list@size@\@tempa\endcsname
928 \ifMT@inlist@
929 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size `~\MT@size' in list}%
930     \MT@dotrue
931     \else
932 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size `~\MT@size' not in list}%
933     \MT@dofalse
934     \expandafter\MT@clist@break
935     \fi
936 }%
937 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size list empty}}%
938 }

```

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

```

939 \def\MT@checklist@font#1{%
940 <!debug> \MT@ifdefined@n@T
941 <debug> \MT@ifdefined@n@TF
942 {MT@#1list@font@\@tempa}%

```

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

```

943 \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
944 \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
945 \@tempb \csname MT@#1list@font@\@tempa\endcsname
946 \ifMT@inlist@
947 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: font `~\MT@font' in list}%
948     \expandafter\MT@clist@break
949     \else
950 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: font `~\MT@font' not in list}%
951     \MT@dofalse
952     \fi
953 }%
954 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: font list empty}}%

```

955 }

### 14.2.1 Protrusion

\MT@protrusion      Set up for protrusion?

956 \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.

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

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

958 \MT@if@list@exists{%

959 \MT@get@font@dimen@six

960 \MT@get@opt

961 \MT@reset@pr@codes

Get the name of the inheritance list and parse it.

962 \MT@get@inh@list

Set an input encoding?

963 \MT@set@inputenc{c}%

Load additional lists?

964 \MT@load@list\MT@pr@c@name

965 \MT@set@listname

Load the main list.

966 \MT@let@cn\@tempc{\MT@pr@c@\MT@pr@c@name}%

967 \expandafter\MT@pr@do\@tempc,\relax,%

968 }\MT@reset@pr@codes

969 }

\MT@gobble@settings

970 \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

971 \def\MT@get@font@dimen@six{%

972 \ifnum\fontdimen6\MT@font=\z@

973 \MT@warning@nl{%

974 Font '\MT@font' does not specify its\MessageBreak

975 \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak

976 \@nameuse{\MT@abbr@\MT@feat} will not work with this font}%

977 \expandafter\MT@gobble@settings

978 \else

979 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%

980 \fi

981 }

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

982 \def\MT@set@all@pr#1#2{%

983 *<debug>*\MT@dinfol{3}{-- lp/rp: setting all to #1/#2}%

984 \let\MT@temp@empty

985 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lcode\MT@font\@tempcnta=#1\relax}}%

986 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rcode\MT@font\@tempcnta=#2\relax}}%

987 \MT@do@font\MT@temp

988 }



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

```

989 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@z@}
990 \let\MT@reset@pr@codes\relax

```

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

```

991 \def\MT@the@pr@code{\@tempcntb}
992 \MT@requires@pdfTeX6{
993   \def\MT@the@pr@code@tr{%
994     \numexpr\@tempcntb+\MT@letterspace@/2\relax
995   }
996 } \relax

```

`\MT@pr@do` Split up the values and set `\lcode` and `\rcode`.

```

997 \def\MT@pr@do#1,{%
998   \ifx\relax#1\@empty\else
999     \MT@pr@split #1==\relax
1000    \expandafter\MT@pr@do
1001    \fi
1002 }

```

`\MT@pr@split` The `keyval` package would remove spaces here, which we needn't do since `\SetProtrusion` ignores spaces in the protrusion list anyway.

```

1003 \def\MT@pr@split#1=#2=#3\relax{%
1004   \def\@tempa{#1}%
1005   \ifx\@tempa\@empty \else
1006     \MT@get@slot
1007     \ifnum\MT@char > \m@ne
1008       \MT@get@char@unit
1009       \MT@pr@split@val#2\relax
1010     \fi
1011   \fi
1012 }

```

`\MT@pr@split@val`

```

1013 \def\MT@pr@split@val#1,#2\relax{%
1014   \def\@tempb{#1}%
1015   \MT@ifempty\@tempb\relax{%
1016     \MT@scale@to@em
1017     \lcode\MT@font\MT@char=\MT@the@pr@code
1018     <debug>\MT@info@n1{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char: [#1]}%
1019   }%
1020   \def\@tempb{#2}%
1021   \MT@ifempty\@tempb\relax{%
1022     \MT@scale@to@em
1023     \rcode\MT@font\MT@char=\MT@the@pr@code
1024     <debug>\MT@info@n1{4}{;;; rp (\MT@char): \number\rcode\MT@font\MT@char: [#2]}%
1025   }%

```

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

```

1026 \MT@ifdefined@c@T\MT@pr@inh@name{%
1027   \MT@ifdefined@n@T\MT@inh@\MT@pr@inh@name @\MT@char @{%
1028     \expandafter\MT@map@tlist@c
1029     \csname MT@inh@\MT@pr@inh@name @\MT@char @\endcsname
1030     \MT@set@pr@heirs
1031   }%
1032 }%

```

1033 }

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

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

```
1034 \MT@maybe@etex{
1035   \MT@requires@pdftex3{
1036     \def\MT@scale@to@em{%
1037       \@tempcntb=\MT@count\relax
```

For really huge fonts (100pt 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`.

```
1038   \MT@scale\@tempcntb \@tempb \MT@dimen@six
1039   \ifnum\@tempcntb=\z@ \else
1040     \MT@scale@factor
1041   \fi
1042 }
```

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

```
1043   \def\MT@get@charwd{%
1044     \MT@count=\fontcharwd\MT@font\MT@char\relax
1045     \setbox\z@=\hbox{\MT@font \char\MT@char}%
1046     \MT@count=\wd\z@
1047     \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1048   }
```

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

```
1049   \MT@requires@pdftex6{
1050     \g@addto@macro\MT@get@charwd{%
1051       \MT@ifdefined@c@T\MT@letterspace@
1052       {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1053     }
1054   }\relax
1055 }
```

No adjustment with versions 0.14f and 0.14g.

```
1056   \def\MT@scale@to@em{%
1057     \MT@count=\@tempb\relax
1058     \ifnum\MT@count=\z@ \else
1059       \MT@scale@factor
1060     \fi
1061   }
```

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

```
1062   \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1063 }
1064 }
```

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

```

1065 \def\MT@get@font@dimen#1{%
1066   \ifnum\fontdimen#1\MT@font=\z@
1067     \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1068       \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1069       You should use a different 'unit' for \MT@curr@list@name}%
1070   \else
1071     \MT@count=\fontdimen#1\MT@font
1072   \fi
1073 }
```

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

```

1074 \MT@maybe@etex{
1075   \def\MT@info@missing@char{%
1076     \MT@info@n1{Character '\the\MT@toks'
1077 X     \iffontchar\MT@font\MT@char
1078       has a width of Opt
1079 X     \else is missing \fi
1080 Q     \MessageBreak (it's probably missing)
1081     \MessageBreak in font '\MT@font'. \MessageBreak
1082     Ignoring protrusion settings for this character}%
1083   }
1084 }
```

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

```

1085 \def\MT@scale@factor{%
1086   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1087     \expandafter\MT@scale\expandafter \@tempcntb
1088     \csname MT@\MT@feat @factor@\endcsname \@m
1089   \fi
1090   \ifnum\@tempcntb>\csname MT@\MT@feat @max@\endcsname\relax
1091     \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @max@\endcsname
1092   \else
1093     \ifnum\@tempcntb<\csname MT@\MT@feat @min@\endcsname\relax
1094       \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @min@\endcsname
1095     \fi
1096   \fi
1097 }
```

`\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.

```

1098 \def\MT@warn@code@too@large#1{%
1099   \@tempcnta=#1\relax
1100   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1101     \expandafter\MT@scale\expandafter \@tempcnta\expandafter
1102     \@m \csname MT@\MT@feat @factor@\endcsname
1103   \fi
1104   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1105   \MT@warning@n1{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
1106     is too large for character \MessageBreak
1107     '\the\MT@toks' in \MT@curr@list@name. \MessageBreak
1108     Setting it to the maximum of \number\@tempcnta}%
1109   \@tempcntb=#1\relax
1110 }
```

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

```

1111 \def\MT@get@opt{%
1112   \MT@set@listname
```

```

\MT@pr@factor@    Apply a factor?
\MT@sp@factor@ 1113 \MT@ifdefined@n@TF{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @factor}{%
1114 \MT@let@nn{MT@MT@feat @factor@}
\MT@kn@factor@ 1115 {MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @factor}%
1116 \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@MT@feat} codes by
1117 \number\csname MT@MT@feat @factor@\endcsname/1000}%
1118 }{%
1119 \MT@let@nn{MT@MT@feat @factor@}{MT@MT@feat @factor}%
1120 }%

\MT@pr@unit@    The unit can only be evaluated here, since it might be font-specific. If it's \@empty,
\MT@sp@unit@    it's relative to character widths, if it's -1, relative to space dimensions.
\MT@kn@unit@ 1121 \MT@ifdefined@n@TF{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @unit}{%
1122 \MT@let@nn{MT@MT@feat @unit@}%
1123 {MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @unit}%
1124 \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\@empty
1125 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
1126 relative to character widths}%
1127 \else
1128 \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\m@ne
1129 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
1130 relative to width of space}%
1131 \fi
1132 \fi
1133 }{%
1134 \MT@let@nn{MT@MT@feat @unit@}{MT@MT@feat @unit}%
1135 }%

\MT@get@space@unit The codes are either relative to character widths, or to a fixed width. For spacing
\MT@get@char@unit and kerning lists, they may also be relative to the width of the interword glue. Only
the setting from the top list will be taken into account.
1136 \let\MT@get@char@unit\relax
1137 \let\MT@get@space@unit\@gobble
1138 \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\@empty
1139 \let\MT@get@char@unit\MT@get@charwd
1140 \else
1141 \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\m@ne
1142 \let\MT@get@space@unit\MT@get@font@dimen
1143 \else
1144 \expandafter\MT@get@unit\csname MT@MT@feat @unit@\endcsname
1145 \fi
1146 \fi

Preset all characters? If so, we surely don't need to reset, too.
1147 \MT@ifdefined@n@TF{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @preset}{%
1148 \csname MT@preset@MT@feat\endcsname
1149 \MT@let@nc{MT@reset@MT@feat @codes}\relax
1150 }%
1151 }

\MT@get@unit@    If unit contains an em or ex, we use the corresponding \fontdimen to obtain the
\MT@get@unit@    real size. Simply converting the em into points might give a wrong result, since
the font probably isn't set up yet, so that these dimensions haven't been updated,
either.
1152 \def\MT@get@unit#1{%
1153 \expandafter\MT@get@unit@#1 e!\@nil
1154 \ifx\x\@empty\else\let#1\x\fi
1155 \@defaultunits\@tempdima#1 pt\relax\@nnil
1156 \ifdim\@tempdima=\z@
1157 \MT@warning@n1{%

```

```

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

```

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

```
1180 \def\MT@set@inputenc#1{%
```

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

```

1181   \def\MT@cat{#1}%
1182   \edef\@tempa{MT@MT@feat @#1\csname MT@MT@feat @#1\name\endcsname @inputenc}%
1183   \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1184 }

```

\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.

```

1185 \MT@addto@setup{%
1186   \@ifpackageloaded{inputenc}{
1187     \@ifpackagelater{inputenc}{2006/02/22}{
1188       \def\MT@set@inputenc@{%
1189         \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1190         \MT@load@inputenc
1191       }
1192     }{
1193       \let\MT@set@inputenc@\MT@load@inputenc
1194     }
1195   }{
1196     \def\MT@set@inputenc@{%
1197       \MT@warning@n1{Key 'inputenc' used in \MT@curr@list@name, but the 'inputenc'
1198       \MessageBreak package isn't loaded. Ignoring input encoding}%
1199     }
1200   }
1201 }

```

\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.

```

1202 \def\MT@load@inputenc{%
1203   \MT@cfg@catcodes
1204   <debug>\MT@edinfo@n1{loading input encoding: \@nameuse{\@tempa}}%
1205   \inputencoding{\@nameuse{\@tempa}}%
1206 }

```

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

```

1207 \def\MT@set@pr@heirs#1{%
1208   \lcode\MT@font#1=\lcode\MT@font\MT@char
1209   \rcode\MT@font#1=\rcode\MT@font\MT@char
1210 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1211 <debug>\MT@info@n1{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
1212 <debug>                                     \number\rcode\MT@font\MT@char}%
1213 }

```

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

```

\MT@preset@pr@ 1214 \def\MT@preset@pr{%
1215   \expandafter\expandafter\expandafter\MT@preset@pr@
1216   \csname MT@pr@c\MT@pr@c@name @preset\endcsname\@nil
1217 }
1218 \def\MT@preset@pr@#1,#2\@nil{%
1219   \ifx\MT@pr@unit@empty
1220     \MT@warn@preset@twidth{pr}%
1221     \let\MT@preset@aux\MT@preset@aux@factor
1222   \else
1223     \def\MT@preset@aux{\MT@preset@aux@space2}%
1224   \fi
1225   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1226   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1227   \MT@set@all@pr\@tempa\@tempb
1228 }

```

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

```

\MT@preset@aux@factor 1229 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 1230   \@tempcntb=#1\relax
1231   \MT@scale@factor
1232   \edef#2{\number\@tempcntb}%
1233 }
1234 \def\MT@preset@aux@space#1#2#3{%
1235   \def\@tempb{#2}%
1236   \MT@get@space@unit#1%
1237   \MT@scale@to@em
1238   \edef#3{\number\@tempcntb}%
1239 }

```

\MT@warn@preset@twidth

```

1240 \def\MT@warn@preset@twidth#1{%
1241   \MT@warning@n1{%
1242     Cannot preset characters relative to their widths\MessageBreak
1243     for \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1244     \MessageBreak relative to lem instead}%
1245 }

```

### 14.2.2 Expansion

\MT@expansion      Set up for expansion?

```

1246 \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).

```

1247 \def\MT@set@ex@codes@s{%
1248   \MT@if@list@exists{%
1249     \MT@get@ex@opt
1250     \MT@reset@ef@codes
1251     \MT@get@inh@list

```

```

1252 \MT@set@inputenc{c}%
1253 \MT@load@list\MT@ex@cc@name
1254 \MT@set@listname
1255 \MT@let@cn\@tempc{MT@ex@cc@MT@ex@cc@name}%
1256 \expandafter\MT@ex@do\@tempc,\relax,%
1257 \MT@expandfont
1258 }\relax
1259 }

```

`\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.

```

1260 \newif\ifMT@nonselected
1261 \def\MT@set@ex@codes@n{%
1262 \MT@nonselectedtrue
1263 \MT@if@list@exists
1264 \MT@get@ex@opt
1265 {%
1266 \let\MT@stretch@MT@stretch
1267 \let\MT@shrink@MT@shrink
1268 \let\MT@step@MT@step
1269 \let\MT@auto@MT@auto
1270 \let\MT@ex@factor@MT@ex@factor
1271}%
1272 \MT@reset@ef@codes
1273 \MT@expandfont
1274 \MT@nonselectedfalse
1275 }

```

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

```
1276 \let\MT@set@ex@codes\MT@set@ex@codes@n
```

`\MT@expandfont` Expand the font.

```

1277 \def\MT@expandfont{%
1278 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1279 }

```

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

```

\MT@reset@ef@codes@
1280 \def\MT@set@all@ex#1{%
1281 <debug>\MT@info@n1{3}{-- ex: setting all to \number#1}%
1282 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1283 }
1284 \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.

```

1285 \MT@requires@pdftex4{
1286 \def\MT@reset@ef@codes{%
1287 \ifnum\MT@ex@factor@=\@m \else
1288 \MT@reset@ef@codes@
1289 \fi
1290 }
1291 }{
1292 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1293 }

```

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

```

1294 \def\MT@ex@do#1,{%
1295 \ifx\relax#1\empty \else

```

```

1296 \MT@ex@split #1==\relax
1297 \expandafter\MT@ex@do
1298 \fi
1299 }

```

\MT@ex@split

```

1300 \def\MT@ex@split#1=#2=#3\relax{%
1301 \def\@tempa{#1}%
1302 \ifx\@tempa\empty \else
1303 \MT@get@slot
1304 \ifnum\MT@char > \m@ne
1305 \@tempcntb=#2\relax

```

Take an optional factor into account.

```

1306 \ifnum\MT@ex@factor@=\@m \else
1307 \MT@scale\@tempcntb \MT@ex@factor@ \@m
1308 \fi
1309 \ifnum\@tempcntb > \MT@ex@max
1310 \MT@warn@ex@too@large\MT@ex@max
1311 \else
1312 \ifnum\@tempcntb < \MT@ex@min
1313 \MT@warn@ex@too@large\MT@ex@min
1314 \fi
1315 \fi
1316 \efcode\MT@font\MT@char=\@tempcntb
1317 <debug>\MT@info@nl{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#2]}%

```

Heirs, heirs, I love thy heirs.

```

1318 \MT@ifdefined@c@T\MT@ex@inh@name{%
1319 \MT@ifdefined@n@T{MT@inh@MT@ex@inh@name @\MT@char @}{%
1320 \expandafter\MT@map@tlist@c
1321 \csname MT@inh@MT@ex@inh@name @\MT@char @\endcsname
1322 \MT@set@ex@heirs
1323 }%
1324 }%
1325 \fi
1326 \fi
1327 }

```

\MT@warn@ex@too@large

```

1328 \def\MT@warn@ex@too@large#1{%
1329 \MT@warning@nl{Expansion factor \number\@tempcntb\space too large for
1330 character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1331 Setting it to the maximum of \number#1}%
1332 \@tempcntb=#1\relax
1333 }

```

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

```

\MT@ex@factor@ 1334 \def\MT@get@ex@opt{%
\MT@stretch@ 1335 \MT@set@listname
1336 \MT@ifdefined@n@TF{MT@ex@cc\MT@ex@cc@name @factor}{%
\MT@shrink@ 1337 \MT@let@cn\MT@ex@factor@{MT@ex@cc\MT@ex@cc@name @factor}%
\MT@step@ 1338 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
1339 }%
\MT@auto@ 1340 \let\MT@ex@factor@\MT@ex@factor
1341 }%
1342 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1343 \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1344 \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
1345 \def\@tempa{autoexpand}%
1346 \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1347 \MT@ifdefined@n@T{MT@ex@cc\MT@ex@cc@name @preset}{%

```



```

1348 \MT@preset@ex
1349 \let\MT@reset@ef@codes\relax
1350 }%
1351 }

\MT@get@ex@opt@
1352 \def\MT@get@ex@opt@#1#2{%
1353 \MT@ifdefined@n@TF{MT@ex@cc@MT@ex@cc@name @#1}{%
1354 \MT@let@nn{MT@#1@}{MT@ex@cc@MT@ex@cc@name @#1}%
1355 \MT@vinfo{... : #2}%
1356 }{%
1357 \MT@let@nn{MT@#1@}{MT@#1}%
1358 }%
1359 }

\MT@set@ex@heirs
1360 \def\MT@set@ex@heirs#1{%
1361 \efcode\MT@font#1=\efcode\MT@font\MT@char
1362 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1363 <debug>\MT@info@n1{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1364 }

\MT@preset@ex
1365 \def\MT@preset@ex{%
1366 \@tempcntb=\csname MT@ex@cc@MT@ex@cc@name @preset\endcsname\relax
1367 \MT@scale@factor
1368 \MT@set@all@ex\@tempcntb
1369 }

```

### 14.2.3 Interword Spacing (Glue)

```

\MT@spacing Adjustment of interword spacing?
1370 \MT@requires@pdftex6{
1371 \def\MT@spacing{\MT@maybe@do{sp}}

\MT@set@sp@codes This is all the same.
1372 \def\MT@set@sp@codes{%
1373 \MT@if@list@exists{%
1374 \MT@get@font@dimen@six
1375 \MT@get@opt
1376 \MT@reset@sp@codes
1377 \MT@get@inh@list
1378 \MT@set@inputenc{c}%
1379 \MT@load@list\MT@sp@cc@name
1380 \MT@set@listname
1381 \MT@let@cn\@tempc{MT@sp@cc@MT@sp@cc@name}%
1382 \expandafter\MT@sp@do\@tempc,\relax,%
1383 }\MT@reset@sp@codes
1384 }

\MT@sp@do
1385 \def\MT@sp@do#1,{%
1386 \ifx\relax#1\empty \else
1387 \MT@sp@split #1=\relax
1388 \expandafter\MT@sp@do
1389 \fi
1390 }

\MT@sp@split
1391 \def\MT@sp@split#1=#2=#3\relax{%

```

```

1392 \def\@tempa{#1}%
1393 \ifx\@tempa\@empty \else
1394 \MT@get@slot
1395 \ifnum\MT@char > \m@ne
1396 \MT@get@char@unit
1397 \MT@sp@split@val#2\relax
1398 \fi
1399 \fi
1400 }

```

\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).

```

1401 \def\MT@sp@split@val#1,#2,#3\relax{%
1402 \def\@tempb{#1}%
1403 \MT@ifempty\@tempb\relax{%
1404 \MT@get@space@unit2%
1405 \MT@scale@to@em
1406 \knbscode\MT@font\MT@char=\@tempcntb
1407 <debug>\MT@info@nl{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1408 }%
1409 \def\@tempb{#2}%
1410 \MT@ifempty\@tempb\relax{%
1411 \MT@get@space@unit3%
1412 \MT@scale@to@em
1413 \stbscode\MT@font\MT@char=\@tempcntb
1414 <debug>\MT@info@nl{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1415 }%
1416 \def\@tempb{#3}%
1417 \MT@ifempty\@tempb\relax{%
1418 \MT@get@space@unit4%
1419 \MT@scale@to@em
1420 \shbscode\MT@font\MT@char=\@tempcntb
1421 <debug>\MT@info@nl{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1422 }%
1423 \MT@ifdefined@c@T\MT@sp@inh@name{%
1424 \MT@ifdefined@n@T\MT@inh@\MT@sp@inh@name @\MT@char @}%
1425 \expandafter\MT@map@tlist@c
1426 \csname MT@inh@\MT@sp@inh@name @\MT@char @\endcsname
1427 \MT@set@sp@heirs
1428 }%
1429 }%
1430 }

```

\MT@set@sp@heirs

```

1431 \def\MT@set@sp@heirs#1{%
1432 \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1433 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1434 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1435 <debug>\MT@info@nl{2}{-- heir of \MT@char: #1}%
1436 <debug>\MT@info@nl{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1437 <debug> \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1438 }

```

\MT@set@all@sp

```

\MT@reset@sp@codes 1439 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1440 <debug>\MT@info@nl{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1441 \let\MT@temp\@empty
1442 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
1443 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1444 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1445 \MT@do@font\MT@temp
1446 }

```

```

1447 \def\MT@reset@sp@codes@{\MT@set@all@sp@z@z@z@}
1448 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1449 \def\MT@preset@sp{%
1450   \expandafter\expandafter\expandafter\MT@preset@sp@
1451   \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1452 }
1453 \def\MT@preset@sp@#1,#2,#3\@nil{%
1454   \ifx\MT@sp@unit@\empty
1455     \MT@warn@preset@tewidth{sp}%
1456     \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
1457     \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
1458     \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
1459   \else
1460     \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
1461     \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
1462     \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
1463   \fi
1464   \MT@set@all@sp@\@tempa\@tempc\@tempb
1465 }
1466 }\relax

```

#### 14.2.4 Additional Kerning

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

```

1467 \MT@requires@pdftex6{
1468 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

1469 \def\MT@set@kn@codes{%
1470   \MT@if@list@exists{%
1471     \MT@get@font@dimen@six
1472     \MT@get@opt
1473     \MT@reset@kn@codes
1474     \MT@get@inh@list
1475     \MT@set@inputenc{c}%
1476     \MT@load@list\MT@kn@c@name
1477     \MT@set@listname
1478     \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
1479     \expandafter\MT@kn@do\@tempc,\relax,%
1480   }\MT@reset@kn@codes
1481 }

```

\MT@kn@do

```

1482 \def\MT@kn@do#1,{%
1483   \ifx\relax#1\@empty \else
1484     \MT@kn@split #1=\relax
1485     \expandafter\MT@kn@do
1486   \fi
1487 }

```

\MT@kn@split

```

1488 \def\MT@kn@split#1=#2=#3\relax{%
1489   \def\@tempa{#1}%
1490   \ifx\@tempa\@empty \else
1491     \MT@get@slot
1492     \ifnum\MT@char > \m@ne
1493       \MT@get@char@unit
1494       \MT@kn@split@val#2\relax
1495     \fi

```

```

1496 \fi
1497 }

```

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

```

1498 \def\MT@kn@split@val#1,#2\relax{%
1499   \def\@tempb{#1}%
1500   \MT@ifempty\@tempb\relax{%
1501     \MT@get@space@unit2%
1502     \MT@scale@to@em
1503     \knbcode\MT@font\MT@char=\@tempcntb
1504   <debug>\MT@info@nl{4}{;;; knbc (\MT@char): \number\knbcode\MT@font\MT@char: [#1]}%
1505   }%
1506   \def\@tempb{#2}%
1507   \MT@ifempty\@tempb\relax{%
1508     \MT@get@space@unit2%
1509     \MT@scale@to@em
1510     \knacode\MT@font\MT@char=\@tempcntb
1511   <debug>\MT@info@nl{4}{;;; knac (\MT@char): \number\knacode\MT@font\MT@char: [#2]}%
1512   }%
1513   \MT@ifdefined@c@T\MT@kn@inh@name{%
1514     \MT@ifdefined@n@T\MT@inh@MT@kn@inh@name @\MT@char @}%
1515     \expandafter\MT@map@tlist@c
1516     \csname MT@inh@MT@kn@inh@name @\MT@char @\endcsname
1517     \MT@set@kn@heirs
1518   }%
1519   }%
1520 }

```

\MT@set@kn@heirs

```

1521 \def\MT@set@kn@heirs#1{%
1522   \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1523   \knacode\MT@font#1=\knacode\MT@font\MT@char
1524   <debug>\MT@info@nl{2}{-- heir of \MT@char: #1}%
1525   <debug>\MT@info@nl{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1526   <debug>                                     \number\knacode\MT@font\MT@char}%
1527 }

```

\MT@set@all@kn

```

\MT@reset@kn@codes 1528 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1529   <debug>\MT@info@nl{3}{-- knac/knbc: setting all to #1/#2}%
1530   \let\MT@temp\@empty
1531   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbcode\MT@font\@tempcnta=#1\relax}}%
1532   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knacode\MT@font\@tempcnta=#2\relax}}%
1533   \MT@do@font\MT@temp
1534 }
1535 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1536 \let\MT@reset@kn@codes\relax

```

\MT@preset@kn

```

\MT@preset@kn@ 1537 \def\MT@preset@kn{%
1538   \expandafter\expandafter\expandafter\MT@preset@kn@
1539   \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1540 }
1541 \def\MT@preset@kn@#1,#2\@nil{%
1542   \ifx\MT@kn@unit@\@empty
1543     \MT@warn@preset@to@width{kn}%
1544     \let\MT@preset@aux\MT@preset@aux@factor
1545   \else
1546     \def\MT@preset@aux{\MT@preset@aux@space2}%
1547   \fi
1548   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1549   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%

```

```

1550 \MT@set@all@kn\@tempa\@tempb
1551 }
1552 }\relax

```

## 14.2.5 Tracking

This only works with pdf<sub>T</sub><sub>E</sub>X 1.40.

```

1553 \MT@requires@pdftex6{

```

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

`\MT@tracking@`

```

\MT@tr@font@list 1554 \let\MT@tr@font@list\@empty
1555 \def\MT@tracking@{%
1556 \MT@exp@one@n\MT@in@cl@list\MT@font\MT@tr@font@list
1557 \ifMT@in@list@else
1558 \MT@maybe@do{tr}%
1559 \ifMT@do@else
1560 \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1561 \fi
1562 \fi
1563 }
1564 \let\MT@tracking\MT@tracking@
1565 \<package>
1566 \<letterspace>\let\MT@tracking\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.

```

1567 \def\MT@set@tr@codes{%
1568 \<package>
1569 \MT@vinfo{Tracking font `'\MT@font'\on@line}%
1570 \MT@get@font@dimen@six
1571 \MT@if@list@exists
1572 \MT@get@tr@opt
1573 \relax
1574 \<package>
1575 \MT@ifdefined@c@TF\MT@letterspace@relax{\let\MT@letterspace@ \MT@letterspace}%
1576 \ifnum\MT@letterspace@=z@else
1577 \<package> \MT@vinfo{... Tracking by \number\MT@letterspace@}%

```

Letterspacing only works in PDF mode.

```

1578 \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.

```

1579 \xdef\MT@lsfont{\csname\expandafter\string\font@name
1580 \number\MT@letterspace@ls\endcsname}%
1581 \expandafter\ifx\MT@lsfont\relax
1582 \<debug>\MT@info@nl{1}{... new letterspacing instance}%

```

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

```

1583 \MT@get@ls@basefont
1584 \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@

```

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

```

1585 \<package>

```

```

1586 \MT@ifdefined@c@TF\MT@tr@ispace
1587 {\let\@tempa\MT@tr@ispace}%
1588 {\edef\@tempa{\MT@letterspace@*,,}}%
1589 \MT@ifdefined@c@TF\MT@tr@ospace
1590 {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1591 {\edef\@tempa{\@tempa,,}}%
1592 \expandafter\MT@tr@set@space\@tempa,%

```

Adjust outer kerning.

```

1593 \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,,}}%
1594 \expandafter\MT@tr@set@okern\@tempa,%
1595 </package>
1596 <*letterspace>
1597 \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1598 * \fontdimen2\MT@lsfont/1000\relax
1599 </letterspace>

```

Disable ligatures (not configurable in letterspace).

```

1600 <package> \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1601 <letterspace> \tagcode\MT@lsfont~f=\m@ne
1602 \fi

```

Finally, let the letterspaced font propagate.

```

1603 \aftergroup\MT@set@lsfont
1604 <package> \let\MT@font\MT@lsfont

```

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

```

\MT@curr@ls 1605 \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
1606 \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.

```

1607 <package>
1608 \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
1609 \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1610 \aftergroup\aftergroup\aftergroup\MT@set@curr@os
1611 \MT@tr@outer@l
1612 </package>

```

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

```

1613 \ifx\MT@ls@adjust\@empty
1614 \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1615 \MT@ls@outer@k
1616 <*letterspace>
1617 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1618 \aftergroup\aftergroup\aftergroup\MT@set@curr@ok
1619 \aftergroup\aftergroup\aftergroup\MT@ls@outer@k
1620 </letterspace>

```

Otherwise, get the current outer kerning and adjust it, for left and right side.

```

1621 <*package>
1622 \else
1623 \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
1624 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1625 \ifdim\MT@outer@kern=z\else \MT@ls@outer@k \fi
1626 \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
1627 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1628 </package>
1629 \fi
1630 <*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).

```
1631 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1632 \aftergroup\aftergroup\aftergroup\MT@set@curr@ok
1633 \aftergroup\aftergroup\aftergroup\MT@tr@outer@r
```

Adjust protrusion values, now and maybe later.

```
1634 <debug>\MT@dinfo@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%
1635 </package>
1636 \MT@do@font{\lcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
1637 \rcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
1638 <package> \let\MT@the@pr@code\MT@the@pr@code@tr
1639 \fi
1640 }
```

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

```
1641 <*package>
1642 \def\MT@get@tr@opt{%
1643 \MT@set@listname
1644 \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name}{%
1645 \MT@let@cn\MT@letterspace{MT@tr@c@\MT@tr@c@name}%
```

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

```
1646 \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @unit}{%
1647 \MT@let@cn\MT@tr@unit@{MT@tr@c@\MT@tr@c@name @unit}%
1648 \ifdim\MT@tr@unit@=1em
1649 \let\MT@tr@unit@\undefined
1650 \else
1651 \MT@let@cn\@tempb{MT@tr@c@\MT@tr@c@name}%
1652 \MT@get@unit\MT@tr@unit@
1653 \let\MT@tr@factor@\@m
1654 \MT@scale@to@em
1655 \edef\MT@letterspace{\number\@tempcntb}%
1656 \fi
1657 }%
1658 }%
```

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

```
\MT@tr@ospace 1659 \MT@get@tr@opt@{spacing} {ispace}%
1660 \MT@get@tr@opt@{outerspacing}{ospace}%
```

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

```
1661 \MT@get@tr@opt@{outerkerning}{okern}%
```

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

```
1662 \MT@get@tr@opt@{noligatures} {ligatures}%
1663 }
```

`\MT@get@tr@opt@`

```
1664 \def\MT@get@tr@opt@#1#2{%
1665 \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @#1}%
1666 {\MT@let@nn{MT@tr@#2}{MT@tr@c@\MT@tr@c@name @#1}}%
1667 }
1668 </package>
```

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

```
1669 <plain>\ifnum\MT@plain=0
1670 \def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}
```

`\lstyle` Disable the tests whether the font should be letterspaced, then trigger the setup.

```
1671 \DeclareRobustCommand\lstyle{%
```

```

1672 <package> \def\MT@feat{tr}%
1673 \let\MT@tracking\MT@set@tr@codes
1674 \selectfont
1675 }

```

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

```

1676 <*plain>
1677 \else
1678 \def\MT@set@lsfont{\MT@lsfont}
1679 \renewcommand\lsstyle{%
1680 \begingroup
1681 \escapechar\m@ne
1682 \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1683 \MT@set@tr@codes
1684 \endgroup
1685 }
1686 \let\textls\undefined
1687 \let\lslig\undefined
1688 \fi
1689 </plain>

```

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

```

1690 \DeclareRobustCommand\lslig[1]{%
1691 {\MT@ifdefined@c@TF\MT@curr@ls{%
1692 \escapechar\m@ne
1693 \MT@get@ls@basefont
1694 \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1695 \kern\MT@outer@kern
1696 \font@name #1%
1697 \kern\MT@outer@kern%
1698 }{#1}}%
1699 }

```

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

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

```

1700 \def\MT@get@ls@basefont{%
1701 \edef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1702 \expandafter\ifx\MT@ls@basefont\relax
1703 \MT@exp@two@c\MT@gl@et\MT@ls@basefont\font@name
1704 \else
1705 <debug>\MT@din@fo@n1{1}{... fixing base font}%
1706 \MT@exp@two@c\let\font@name\MT@ls@basefont
1707 \fi
1708 }

```

`\MT@tr@noligatures` pdf<sub>T</sub><sub>E</sub>X 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.

```

1709 <*package>
1710 \MT@requires@pdftex7{
1711 \def\MT@tr@noligatures{%
1712 \ifx\MT@tr@ligatures\empty
1713 \MT@noligatures@\MT@lsfont\undefined
1714 \else
1715 \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1716 \fi
1717 }
1718 }{
1719 \def\MT@tr@noligatures{%

```



```

1720 \MT@warning@n1{%
1721     Disabling selected ligatures is only possible since\MessageBreak
1722     pdftex 1.40.4. Disabling all ligatures instead}%
1723 \global\let\MT@tr@noligatures\relax
1724 }
1725 }

```

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

```
1726 \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.

```

1727 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
1728 <debug>\MT@info@n12{... orig. space:
1729 <debug> \the\fontdimen2\MT@lsfont, \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
1730 <debug> \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
1731 \let\MT@temp\@empty
1732 \MT@tr@set@space@{#1}{#4}{2}\@empty
1733 \MT@tr@set@space@{#2}{#5}{3}\@plus
1734 \MT@tr@set@space@{#3}{#6}{4}\@minus
1735 \global\MT@let@nc\MT@outer@space\expandafter\string\font@name\MT@temp
1736 <debug>\MT@info@n12{... inner space:
1737 <debug> \the\fontdimen2\MT@lsfont, \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
1738 <debug>\MT@info@n12{... outer space: \MT@temp}%
1739 }

```

\MT@tr@set@space@     If outer spacing settings don't exist, they will be inherited from the inner spacing settings.

```

1740 \def\MT@tr@set@space@#1#2#3#4{%
1741     \MT@ifempty{#2}{%
1742         \MT@ifempty{#1}{%
1743             \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
1744         }{%
1745             \MT@tr@set@space@@{#1}{#3}{1000}%
1746             \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1747             \fontdimen#3\MT@lsfont=\@tempdima
1748         }%
1749     }{%
1750         \MT@tr@set@space@@{#2}{#3}{2000}%
1751         \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1752         \MT@ifempty{#1}\relax{%
1753             \MT@tr@set@space@@{#1}{#3}{1000}%
1754             \fontdimen#3\MT@lsfont=\@tempdima
1755         }%
1756     }%
1757 }

```

\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.

```

1758 \def\MT@tr@set@space@@#1#2#3{%
1759     \MT@test@ast#1*\@nil{%
1760         \MT@ifdefined@cTF\MT@tr@unit@
1761         {\edef\@tempb{#1}\MT@scale@to@em}
1762         {\@tempcntb=#1\relax}%
1763         \@tempdima=\dimexpr \dimexpr \@tempcntb sp*\MT@dimen@six/1000\relax
1764         -\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).

```
1765 \ifnum#2=\tw@
```

```

1766     \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
1767     \fi
1768     \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
1769   }{%
1770     \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
1771     \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
1772   }%
1773   <debug>\MT@info{n13{... : font dimen #2 (#1): \the\@tempdima}%
1774 }

```

\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).

```

1775 \def\MT@tr@outer@l{%
1776   \ifdim\lastskip>5sp
1777     \edef\x{\the\lastskip plus 0pt}%
1778     \setbox\z@\hbox{\MT@outer@space=\x}%
1779     \ifdim\wd\z@>\z@
1780   <debug>\MT@info2{((( adjusting pre spaces: \the\MT@outer@space}%
1781     \unskip \hspace\MT@outer@space\relax

```

Disable left outer kerning.

```

1782     \let\MT@ls@outer@k\relax
1783   \fi
1784 \fi
1785 }

```

\MT@tr@outer@r 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@next 1786 \def\MT@tr@outer@r{%
1787   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
1788 }
1789 \def\MT@tr@outer@r@{%
1790   \def\MT@temp*#1{\hspace\MT@outer@space
1791 <debug>\MT@info2{((( adjusting post spaces (1): \the\MT@outer@space}%
1792   }%
1793   \ifcat\egroup\noexpand\MT@tr@outer@next
1794     \unkern\egroup
1795     \MT@set@curr@ok
1796     \MT@set@curr@os
1797     \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\x=}%
1798   \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.

```

1799     \ifx\maybe@ic\MT@tr@outer@next
1800       \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\x=}%
1801     \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).

```

1802     \ifx\check@icr\MT@tr@outer@next
1803       \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\x=}%
1804     \else
1805       \ifx\@sptoken\MT@tr@outer@next
1806         \def\MT@temp* {\hspace\MT@outer@space
1807 <debug>\MT@info2{((( adjusting post spaces (2): \the\MT@outer@space}%
1808         }%
1809       \else
1810         \ifx-\MT@tr@outer@next

```

```

1811         \def\MT@temp*~{\nobreak\hskip\MT@outer@space
1812 <debug>\MT@edinfo2{}}) adjusting post spaces (3): \the\MT@outer@space}%
1813     }%
1814     \else
1815         \ifx\MT@tr@outer@next \else
1816             \ifx\space\MT@tr@outer@next \else
1817                 \ifx\@xobeysp\MT@tr@outer@next \else

```

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

```

1818         \def\MT@temp*{\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k\fi}%
1819         \let\MT@tr@outer@next\relax
1820     \fi\fi\fi\fi\fi\fi\fi\fi
1821     \MT@temp*%
1822 }

```

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

```

\MT@tr@outer@icr@ 1823 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
1824 \def\MT@tr@outer@icr@{%
1825     \let\@let@token= \MT@tr@outer@next
1826     \maybe@ic@
1827 }

```

For older pdf<sub>T</sub><sub>E</sub>X versions, let it relax.

```

1828 }{
1829     \DeclareRobustCommand\lsstyle{%
1830         \MT@warning{Letterspacing only works with pdftex version 1.40\MessageBreak
1831             or newer. You might want to use the `soul' package\MessageBreak
1832             instead}%
1833         \MT@glet\lsstyle\relax
1834     }
1835 }
1836 </package>

```

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

```

1837 \DeclareRobustCommand\textls{%
1838     \hmode\bgroup
1839     \@ifstar{\let\MT@ls@adjust@empty\MT@textls}%
1840     {\let\MT@ls@adjust\relax \MT@textls}%
1841 }

```

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

```

\MT@letterspace@ 1842 \newcommand\MT@textls[2][{}]{%
1843     \MT@ifempty{#1}%
1844     {\let\MT@letterspace@\@undefined}%
1845     {\KV@esp@def\MT@letterspace@{#1}%
1846         \MT@ls@too@large\MT@letterspace@}%
1847     \lsstyle #2%
1848     \egroup
1849 }

```

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

```

1850 \def\MT@ls@too@large#1{%
1851     \ifnum#1>\MT@tr@max
1852         \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
1853         \let#1\MT@tr@max
1854     \else
1855         \ifnum#1<\MT@tr@min
1856             \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
1857             \let#1\MT@tr@min
1858         \fi
1859     \fi
1860 }

```

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

`\MT@tr@set@okern`

```

1861 \newdimen\MT@outer@kern
1862 <package>
1863 \def\MT@tr@set@okern#1,#2,{%
1864   \let\MT@temp\@empty
1865   \MT@tr@set@okern@{#1}%
1866   \MT@tr@set@okern@{#2}%
1867   \global\MT@let@nc\MT@outer@kern\expandafter\string\font@name}\MT@temp
1868 <debug>\MT@diminfo@n12{... outer kerning:
1869 <debug>   (#1,#2) = \nameuse{MT@outer@kern\expandafter\string\font@name}}%
1870 }

\MT@tr@set@okern@
1871 \def\MT@tr@set@okern@#1{%
1872   \MT@test@ast#1*\@nil{%
1873     \MT@ifdefined@c@TF\MT@tr@unit@
1874     {\edef\@tempb{#1}\MT@scale@to@em}
1875     {\@tempcntb=#1\relax}%
1876     \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
1877   }%
1878   \MT@ifempty\@tempa{\let\@tempa\@m}\relax
1879   \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
1880     * \fontdimen6\MT@lsfont/2000\relax
1881   }%
1882   \advance\@tempdima -\dimexpr \MT@letterspace@ sp
1883     * \fontdimen6\MT@lsfont/2000\relax
1884   \edef\MT@temp{\MT@temp{\the\@tempdima}}%
1885 }
1886 </package>

\MT@ls@outer@k Adjust outer kerning.
1887 \def\MT@ls@outer@k{\ifhmode\kern\MT@outer@kern\relax\fi}
1888 <package>

```

### 14.2.6 Disabling Ligatures

`\MT@noligatures` The possibility to disable ligatures is a new features of pdf<sub>T</sub><sub>E</sub>X 1.30.

```

1889 \MT@requires@pdftex5{
1890   \def\MT@noligatures{%
1891     \MT@dotrue
1892     \let\@tempa\MT@n1@setname
1893     \MT@map@c@list@n{font,encoding,family,series,shape,size}{%
1894       \MT@ifdefined@n@TF\MT@checklist@#1}%
1895       {\csname MT@checklist@#1\endcsname}%
1896       {\MT@checklist@{#1}}}%
1897     {n1}%
1898   }%
1899   \ifMT@do
1900     \MT@noligatures@\MT@font\MT@n1@ligatures
1901   \fi
1902 }

\MT@noligatures@ This is also used by \MT@set@tr@codes.
1903 \def\MT@noligatures@#1#2{%
1904   \MT@ifdefined@c@TF#2{%

Early MiKTEX versions (before 2.5.2579) didn't know \tagcode.
1905   \MT@ifdefined@c@TF\tagcode{%
1906     \MT@map@c@list@c#2{%

```

```

1907      \KV@esp@def\@tempa{##1}\MT@get@slot
1908      \ifnum\MT@char > \m@ne
1909      \tagcode#1\MT@char=\m@ne\fi
1910    }%
1911    \MT@vinfo{... Disabling ligatures for characters: #2}%
1912  }{%
1913    \pdfnoligatures#1%
1914    \MT@warning{Cannot disable selected ligatures (pdfTeX doesn't\MessageBreak
1915      know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
1916      the font instead}%
1917  }%
1918 }{%
1919   \pdfnoligatures#1%
1920   \MT@vinfo{... Disabling ligatures}%
1921 }%
1922 }
1923 }\relax

```

### 14.2.7 Loading the Configuration

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

```

1924 \def\MT@load@list#1{%
1925   \edef\@tempa{#1}%
1926   \MT@let@cn\@tempb{\MT@MT@feat @c@\@tempa @load}%
1927   \MT@ifstreq\@tempa\@tempb{%
1928     \MT@error{\@nameuse{\MT@abbr@\MT@feat} list `@\@tempa' cannot load itself}{}%
1929   }{%
1930     \ifx\@tempb\relax \else
1931       \MT@ifdefined@n@TF{\MT@\MT@feat @c@\@tempb}{%
1932         \MT@vinfo{... : First loading \@nameuse{\MT@abbr@\MT@feat} list `@\@tempb'}%
1933         \begingroup
1934           \MT@load@list\@tempb
1935         \endgroup
1936         \edef\MT@curr@list@name{\@nameuse{\MT@abbr@\MT@feat} list
1937           \noexpand\MessageBreak`@\@tempb'}%
1938         \MT@let@cn\@tempc{\MT@\MT@feat @c@\@tempb}%
1939         \expandafter\csname \MT@\MT@feat @do\expandafter\endcsname\@tempc,\relax,%
1940       }{%
1941         \MT@error{\@nameuse{\MT@abbr@\MT@feat} list `@\@tempb' undefined.\MessageBreak
1942           Cannot load it from list `@\@tempa'}{}%
1943       }%
1944     \fi
1945   }%
1946 }

```

`\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.

```

1947 \let\MT@file@list\empty
1948 \def\MT@find@file#1{%
1949   \MT@in@clist{#1}\MT@file@list
1950   \ifMT@inlist@ \else

```

Check for existence of the file only once.

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.

```

1951   \MT@begin@catcodes
1952   \let\MT@begin@catcodes\relax
1953   \let\MT@end@catcodes\relax
1954   \InputIfFileExists{mt-#1.cfg}{%

```

```

1955 \edef\MT@curr@file{mt-#1.cfg}%
1956 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1957 \MT@xadd\MT@file@list{#1,}%
1958 }{%
1959 \MT@get@basefamily#1\@empty\@empty\@empty\@nil
1960 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1961 \ifMT@inlist@
1962 \MT@xadd\MT@file@list{#1,}%
1963 \else
1964 \InputIfFileExists{mt-\@tempa.cfg}{%
1965 \edef\MT@curr@file{mt-\@tempa.cfg}%
1966 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1967 \MT@xadd\MT@file@list{\@tempa,#1,}%
1968 }{%
1969 \MT@vinfo{... No configuration file mt-#1.cfg}%
1970 \MT@xadd\MT@file@list{#1,}%
1971 }%
1972 \fi
1973 }%
1974 \endgroup
1975 \fi
1976 }

```

`\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.

```

1977 \def\MT@cfg@catcodes{%
1978 \makeatletter
1979 \catcode\^7%
1980 \catcode\ 9%
1981 \catcode\^^I9%
1982 \catcode\^^M9%
1983 \catcode\\\z@
1984 \catcode\{\@ne
1985 \catcode\}\tw@
1986 \catcode\#6%
1987 \catcode\%14%
1988 \MT@map@tlist@n
1989 {\!\"$&'\"(\)*+,\-\.\/\:\;\<=\>?\[\]\_\"|\~}%
1990 \makeoother
1991 }

```

`\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.

```

1992 \def\MT@begin@catcodes{%
1993 \begingroup
1994 \MT@cfg@catcodes
1995 }

```

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

```

1996 \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

Table 3: 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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

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.

```

1997 \def\MT@get@basefamily#1#2#3#4\@nil{%
1998   \ifx\@empty#4%
1999     \def\@tempa{#1#2#3}%
2000   \else
2001     \let\@tempa\@empty
2002     \edef\@tempb{#1#2#3#4}%
2003     \expandafter\MT@get@basefamily@\@tempb\@nil
2004   \fi
2005 }
```

\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’.

```

2006 \def\MT@get@basefamily@#1#2\@nil{%
2007   \edef\@tempa{\@tempa#1}%
2008   \ifx\#2\@expandafter\@gobble\else\expandafter\@firstofone\fi
2009   {\MT@in@tlist{#2}\MT@variants
2010    \ifMT@in@tlist\else\MT@get@basefamily@#2\@nil\fi}%
2011 }
```

\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@ 2012 \def\MT@get@listname#1{%
2013   <debug>\MT@dinfo@n1{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
2014   \let\MT@listname\@undefined
2015   \def\@tempb{#1}%
2016   \MT@map@tlist@c\MT@try@order\MT@get@listname@
2017 }
2018 \def\MT@get@listname@#1{%
2019   \expandafter\MT@next@listname#1%
2020   \ifx\MT@listname\@undefined \else
2021     \expandafter\MT@tlist@break
2022   \fi
2023 }
```

\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 3 in the documentation part any longer and can cast it off here.

```

2024 \def\MT@try@order{%
2025   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2026   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2027 }
```

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

```

2028 \def\MT@next@listname#1#2#3#4{%
2029   \edef\@tempa{\MT@encoding
2030     /\ifnum#1=\@ne \MT@family\fi
```

```

2031          /\ifnum#2=\@ne \MT@series\fi
2032          /\ifnum#3=\@ne \MT@shape\fi
2033          /\ifnum#4=\@ne *\fi
2034          \MT@context}%
2035 <debug>\MT@info{n}{1}{trying \@tempa}%
2036 \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
2037 \MT@next@listname@#4%
2038 }{%

```

Also try with an alias family.

```

2039 \ifnum#1=\@ne
2040 \ifx\MT@familyalias\@empty \else
2041 \edef\@tempa{\MT@encoding
2042 \MT@familyalias
2043 /\ifnum#2=\@ne \MT@series\fi
2044 /\ifnum#3=\@ne \MT@shape\fi
2045 /\ifnum#4=\@ne *\fi
2046 \MT@context}%
2047 <debug>\MT@info{n}{1}{(alias) \@tempa}%
2048 \MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%
2049 \MT@next@listname@#4%
2050 }%
2051 \fi
2052 \fi
2053 }%
2054 }

```

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

```

2055 \def\MT@next@listname@#1{%
2056 \ifnum#1=\@ne
2057 \expandafter\MT@in@rlist\csname MT@\@tempb @\@tempa @size\endcsname
2058 \ifMT@inlist@
2059 \let\MT@listname\MT@size@name
2060 \fi
2061 \else
2062 \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
2063 \fi
2064 }

```

\MT@if@list@exists

```

\MT@context 2065 \def\MT@if@list@exists{%
2066 \MT@let@cn\MT@context{MT@\MT@feat @context}%
2067 \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
2068 \MT@get@listname{\MT@feat @c}%
2069 \MT@ifdefined@c@TF\MT@listname{%
2070 \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
2071 \ifMT@nonselected
2072 \MT@vinfo{... Applying non-selected expansion (list '\MT@listname')}%
2073 \else
2074 \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list '\MT@listname'}%
2075 \fi
2076 \@firstoftwo
2077 }{%

```

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

```

2078 \MT@let@nc{MT@\MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

2079 \ifMT@nonselected
2080 \MT@vinfo{... Applying non-selected expansion (no list)}%
2081 \else

```



Tracking doesn't require a list, either.

```

2082 \MT@ifstreq\MT@feat{tr}\relax{%
2083 \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
2084 for font\MessageBreak`\MT@font'%
2085 \ifx\MT@context\@empty\else\space(context: `\MT@context')\fi.
2086 Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
2087 }%
2088 \fi
2089 \@secondoftwo
2090 }%
2091 }

```

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

```

\MT@context 2092 \def\MT@get@inh@list{%
2093 \let\MT@context\@empty
2094 \MT@get@listname{\MT@feat @inh}%
2095 \MT@ifdefined@c@TF\MT@listname{%
2096 \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
2097 <debug>\MT@dinfo@n{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
2098 <debug>`\MT@listname'}%
2099 \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%

```

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

```

2100 \ifx\@tempc\@empty \else
2101 <debug>\MT@dinfo@n{1}{parsing inheritance list ...}%

```

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

```

2102 \begingroup
2103 \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\MT@listname'}%
2104 \MT@set@inputenc{inh}%
2105 \expandafter\MT@inh@do\@tempc,\relax,%
2106 \global\MT@let@nc{MT@\MT@feat @inh@\MT@listname}\@empty
2107 \endgroup
2108 \fi
2109 }{%
2110 \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
2111 }%
2112 }

```

#### 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@ 2113 \def\MT@get@slot{%
2114 \escapechar`\
2115 \let\MT@char@m@ne
2116 \MT@norestoretrue

```

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

```

2117 \MT@toks=\expandafter{\@tempa}%

```

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

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

```

2118 \expandafter\MT@is@letter\@tempa\relax\relax
2119 \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.

```

2120 \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.

```

2121 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
2122 \MT@is@symbol

```

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

```

2123 \expandafter\MT@is@composite\@tempa\relax\relax}%
2124 \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.

```

2125 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2126 \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2127 \fi
2128 \fi
2129 \let\MT@char\MT@char@
2130 \ifnum\MT@char < \z@
2131 \MT@warn@unknown
2132 \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.

```

2133 \ifMT@norest \else
2134 \MT@warn@rest
2135 \let\MT@char\m@ne
2136 \fi
2137 \fi
2138 \escapechar\m@ne
2139 }

```

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

```

2140 \newif\ifMT@norest

```

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

```

2141 \def\MT@is@letter#1#2\relax{%
2142 \ifcat a\noexpand#1\relax
2143 \edef\MT@char@\number`#1}%
2144 \ifx\#2\%
2145 <debug>\MT@debug@n1{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2146 \else
2147 \MT@norestfalse
2148 \fi

```

```

2149 \else
2150 \ifcat !\noexpand#1\relax
2151 \edef\MT@char@\number`#1}%
2152 <debug>\MT@info@nl{3}{> `the\MT@toks' is a character (\MT@char@)}%
2153 \ifx\#2\%
2154 \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
2155 \else
2156 \MT@noestfalse
2157 \expandafter\MT@is@number#1#2\relax\relax
2158 \fi
2159 \fi
2160 \fi
2161 }

```

`\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.

```

2162 \def\MT@is@number#1#2#3\relax{%
2163 \ifx\relax#3\relax \else
2164 \ifx\relax#2\relax \else
2165 \MT@noesttrue
2166 \if#1"\relax
2167 \def\x{\uppercase{\edef\MT@char@\number#1#2#3}}\x
2168 <debug>\MT@info@nl{3}{> ... a hexadecimal number: \MT@char@}%
2169 \else
2170 \if#1'\relax
2171 \def\MT@char@\number#1#2#3}%
2172 <debug>\MT@info@nl{3}{> ... an octal number: \MT@char@}%
2173 \else
2174 \MT@ifint{#1#2#3}{%
2175 \def\MT@char@\number#1#2#3}%
2176 <debug>\MT@info@nl{3}{> ... a decimal number: \MT@char@}%
2177 } \MT@noestfalse
2178 \fi
2179 \fi
2180 \ifnum\MT@char@ > \ccclv
2181 \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2182 \let\MT@char@\m@ne
2183 \fi
2184 \fi
2185 \fi
2186 }

```

`\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.

```

2187 \def\MT@is@active#1#2\@nil{%
2188 \ifnum\catcode`#1 = \active
2189 \begingroup
2190 \set@display@protect
2191 \let\IeC\@firstofone
2192 \let\@inpenc@undefined@\MT@undefined@char

```

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

```
2193 \def\UTFviii@defined##1{\ifx ##1\relax
2194 \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
```

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

```
2195 \MT@ifdefined@c@T\PrerenderUnicode
2196 {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
2197 \edef\x{\endgroup
2198 \def\noexpand\@tempa{\@tempa}%
```

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

```
2199 \MT@toks={\the\MT@toks\space(=\@tempa)}%
2200 }%
2201 \x
2202 \fi
2203 }
```

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

```
2204 \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.

```
2205 \def\MT@is@symbol{%
2206 \expandafter\def\expandafter\MT@char\expandafter
2207 {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2208 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2209 \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2210 \ifnum\MT@char@ < \z@
```

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

```
2211 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2212 \fi
2213 }
```

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

```
\MT@charstring 2214 \begingroup
2215 \catcode`\=/\z@
2216 /MT@map@tlist@n{/CHAR}/@makeoother
2217 /lowercase{%
2218 /def/x{/endgroup
2219 /def/MT@charstring{\CHAR}%
2220 /def/MT@is@char##1\CHAR"##2##3##4/relax{%
2221 /ifx/relax##1/relax
2222 /if##3\relax
2223 /edef/MT@char@{/number"##2}%
2224 /MT@ifstreq/MT@charstring{##3##4}/relax/MT@noestfalse
2225 /else
2226 /edef/MT@char@{/number"##2##3}%
2227 /MT@ifstreq/MT@charstring{##4}/relax/MT@noestfalse
2228 /fi
2229 <debug> /MT@dinfo@n1{3}{> `/the/MT@toks' is a \char (/MT@char@)}%
2230 /fi
2231 }%
2232 }%
2233 }
2234 /x
```

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

```
2235 \def\MT@is@composite#1#2\relax{%
2236   \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`.

```
2237   \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2238     \string\csname\MT@encoding\endcsname
2239     \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
2240   \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2241   \fi
2242 }
```

(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 2243 \def\MT@set@listname{%
2244   \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
2245   ~\@nameuse{MT@\MT@feat @c@name}'}%
2246 }
```

`\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.

```
2247 \def\MT@warn@ascii{%
2248   \MT@warning@nl{Character `the\MT@toks' (= \MT@char@)
2249   is outside of ASCII range.\MessageBreak
2250   You must load the `inputenc' package before using\MessageBreak
2251   8-bit characters in \MT@curr@list@name}%
2252 }
```

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

```
2253 \def\MT@warn@number@too@large#1{%
2254   \MT@warning@nl{%
2255     Number #1 in encoding `the\MT@encoding' too large!\MessageBreak
2256     Ignoring it in \MT@curr@list@name}%
2257 }
```

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

```
2258 \def\MT@warn@rest{%
2259   \MT@warning@nl{%
2260     Unknown slot number of character\MessageBreak`the\MT@toks'%
2261     \MT@warn@maybe@inputenc\MessageBreak
2262     in font encoding `\'MT@encoding'.\MessageBreak
2263     Make sure it's a single character\MessageBreak
2264     (or a number) in \MT@curr@list@name}%
2265 }
```

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

```
2266 \def\MT@warn@unknown{%
2267   \MT@warning@nl{%
2268     Unknown slot number of character\MessageBreak`the\MT@toks'%
2269     \MT@warn@maybe@inputenc\MessageBreak
2270     in font encoding `\'MT@encoding' in \MT@curr@list@name}%
2271 }
```

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

```
2272 \def\MT@warn@maybe@inputenc{%
2273   \MT@ifdefined@n@T
2274   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
2275   { (input encoding `\'@nameuse
2276     {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2277 }
```

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

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

In contrast to the `pdfcprot` package, there is no need to declare the fonts in advance that should benefit from micro-typographic treatment. Also, only those fonts that are actually being used will be set up for expansion and protrusion.

For my reference:

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

Up to version 1.3 of this package, we were using `\define@newfont` as the hook, which is only called for *new* fonts, and therefore seemed the natural choice. However, this meant that we had to take special care to catch all fonts: we additionally had to set up the default font, the error font (if it wasn't the default font), we had

to check for some packages that might have been loaded before microtype and were loading fonts, e. g., jurabib, ledmac, pifont (loaded by hyperref), 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). Then I learned that even my favourite class, memoir, loads fonts. 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 2278 \let\MT@font@list\empty
2279 \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`.

```
2280 </package>
2281 <plain>\ifnum\MT@plain=0
2282 \MT@addto@setup{%
```

`\MT@orig@pickupfont` microtype also seems to work with CJK.

```
2283 \ifpackageloaded{CJK}{
2284   \ifpackageafter{CJK}{2006/10/17}
2285   {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}} % 4.7.0
2286   {\def\MT@orig@pickupfont{\ifundefined{CJK@plane}}}}
2287   \g@addto@macro\MT@orig@pickupfont{%
2288     {\expandafter\ifx\font@name\relax\define@newfont\fi}%
2289     {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2290       \define@newfont\else\def\font@name
2291         {\csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}
2292   }{
2293   \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
2294   }
```

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

```
2295 \ifx\pickup@font\MT@orig@pickupfont \else
2296   \MT@warning@nl{%
2297     Command \string\pickup@font\space is not defined as expected.%
2298     \MessageBreak Patching it anyway. Some things may break%
2299   <*package>
2300   .\MessageBreak Double-check whether micro-typography is indeed%
2301   \MessageBreak applied to the document.%
2302   \MessageBreak (Hint: Turn on `verbose' mode)%
2303 </package>
2304   }
2305 \fi
```

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

```
2306 \g@addto@macro\pickup@font{\begingroup}
```

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

```
2307 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
2308 \g@addto@macro\pickup@font{%
2309   \escapechar\m@ne
2310 <*package>
2311 <debug> \global\MT@inannottrue
2312 <debug> \MT@glet\MT@pdf@annot\empty
2313 <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.

```

2314 \MT@let@cn\MT@font{MT@subst@expandafter\string\font@name}%
2315 \ifx\MT@font\relax
2316 \let\MT@font\font@name
2317 \else
2318 \ifx\MT@font\font@name \else
2319 <debug> \MT@addto@annot{= substituted with \MT@font}%
2320 \MT@register@subst@font
2321 \fi
2322 \fi
2323 \MT@setupfont
2324 </package>
2325 <letterspace> \MT@tracking
2326 \endgroup
2327 }
2328 <*package>

```

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

```
2329 \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.

```

2330 \g@addto@macro\do@subst@correction
2331 { \edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2332 \global\MT@let@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, L<sup>A</sup>T<sub>E</sub>X takes care that the fonts used for the `\accent` are already set up, so that we cannot be overlooking them.

```

2333 \let\MT@orig@add@accent\add@accent
2334 \def\add@accent#1#2{%
2335 \let\pickup@font\MT@orig@pickupfont
2336 \MT@orig@add@accent{#1}{#2}%
2337 \let\pickup@font\MT@pickupfont
2338 }
2339 </package>
2340 }
2341 <plain>\fi
2342 <*package>

```

Consequently, we are (if all goes well) 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.

```
2343 \def\MT@check@font{\MT@expone@n\MT@in@clist\MT@font\MT@font@list}
```

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

```
2344 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
```

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

```
2345 \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.

```
2346 \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.

```
2347 \def\MT@check@font@cx{%
2348   \MT@if@true
2349   \MT@map@clist@c\MT@active@features{%
2350     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
2351     \MT@font \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2352     \ifMT@inlist@
2353       \MT@let@nc{\MT@%nameuse{MT@abbr@##1}}\relax
2354     \else
2355       \MT@if@false
2356     \fi
2357   }%
2358   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2359 }
```

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

```
2360 \def\MT@register@subst@font@cx{%
2361   \MT@map@clist@c\MT@active@features{%
2362     \expandafter\MT@xadd
2363     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2364     {\font@name,}%
2365   }%
2366 }
```

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

```
2367 \def\MT@register@font@cx{%
2368   \MT@map@clist@c\MT@active@features{%
2369     \expandafter\ifx\csname MT@%nameuse{MT@abbr@##1}\endcsname\relax\else
2370     \expandafter\MT@xadd
2371     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2372     {\MT@font,}%
2373     \def\@tempa{##1}%
2374     \expandafter\MT@map@tlist@c
2375     \csname MT@##1@doc@contexts\endcsname
2376     \MT@maybe@rem@from@list
2377   \fi
2378   }%
2379 }
```

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

```
2380 \def\MT@maybe@rem@from@list#1{%
2381   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@%tempa @context\endcsname}\relax{%
2382     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
2383     \MT@font \csname MT@%tempa @#1font@list\endcsname
2384   }%
2385 }
```

`\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.

```

2386 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
2387 \MT@addto@setup{%
2388   \DeclareRobustCommand\microtypecontext[1]{%
2389     \MT@setup@contexts
2390     \let\MT@reset@context\relax
2391     \setkeys{MTC}{#1}%
2392     \selectfont
2393     \MT@reset@context
2394   }%
2395 }

```

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

```

2396 \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@`

```

2397 \def\MT@reset@context@{%
2398   \MT@vinfo{<<< Resetting contexts\on@line
2399   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
2400   <debug>           /\MT@tr@context/\MT@kn@context/\MT@sp@context
2401   }%
2402   \selectfont
2403 }

```

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

```

2404 \def\MT@setup@contexts{%
2405   \MT@map@clist@c\MT@active@features
2406   {\global\MT@let@nc{\MT@##1@font@list}\MT@font@list}%
2407   \MT@gl@et\MT@check@font\MT@check@font@cx
2408   \MT@gl@et\MT@register@font\MT@register@font@cx
2409   \MT@gl@et\MT@register@subst@font\MT@register@subst@font@cx
2410   \MT@gl@et\MT@setup@contexts\relax
2411 }

```

Define context keys.

```

2412 \MT@map@clist@c\MT@features@long{%
2413   \define@key{MTC}{#1}[]{%
2414     \edef\@tempb{\@nameuse{\MT@rbba#1}}%
2415     \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2416     \ifMT@inlist@

```

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

```

2417   \MT@ifempty{##1}{\def\MT@val{0}}{\def\MT@val{##1}}%
2418   \expandafter\ifx\csname MT@\@tempb @context\endcsname\MT@val
2419   <debug> \MT@vinfo{1}{>>> no change of #1 context: \MT@val}%
2420   \else
2421     \MT@vinfo{>>> Changing #1 context to \MT@val'\MessageBreak\on@line
2422   <debug> \space{previous: \@nameuse{\MT@\@tempb @context}}}%
2423   }%
2424   \def\MT@reset@context{\aftergroup\MT@reset@context@}%

```

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

```

2425   \global\MT@let@nn{\MT@reset@\@tempb @codes}\MT@reset@\@tempb @codes}%

```

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

```

2426   \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
2427   \MT@val \csname MT@\@tempb @doc@contexts\endcsname
2428   \ifMT@inlist@ \else

```

```

2429         \expandafter\MT@add\csname MT@\@tempb @doc@contexts\endcsname{{\MT@val}}}%
2430 <debug> \MT@info{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}}%
2431         \fi
2432         \MT@edef\MT@\@tempb @context{{\MT@val}}%
2433         \fi
2434     \fi
2435 }%
2436 }

```

\MT@pr@context     Initialise the contexts.

\MT@ex@doc@context 2437 \MT@exp@one\MT@map@clist\MT@features,nl}%

\MT@tr@context 2438 \MT@def\MT@#1@context{{@}}%

\MT@sp@context 2439 \MT@def\MT@#1@doc@contexts{{@}}%

\MT@kn@context 2440 }

\MT@kn@context 2441 \let\MT@extra@context\@empty

\MT@pr@doc@contexts

\MT@ex@doc@context

\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.

```

2442 \def\DeclareMicrotypeSet{%
2443     \ifstar
2444         \MT@DeclareSetAndUseIt
2445         \MT@DeclareSet
2446 }

\MT@DeclareSet

2447 \newcommand\MT@DeclareSet[3] [] {%
2448     \KV@sp@def\@tempa{#1}%
2449     \MT@ifempty\@tempa{%
2450         \MT@map@clist\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%
2451     }%
2452     \MT@map@clist\@tempa{%
2453         \KV@sp@def\@tempa{##1}%
2454         \MT@ifempty\@tempa\relax{%
2455             \MT@is@feature{set declaration `#2'}{%
2456                 \MT@exp@one\MT@declare@sets
2457                 {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
2458             }%
2459         }%
2460     }%
2461 }%
2462 }

```

\MT@DeclareSetAndUseIt

```

2463 \newcommand\MT@DeclareSetAndUseIt[3] [] {%
2464     \MT@DeclareSet[#1]{#2}{#3}%
2465     \UseMicrotypeSet[#1]{#2}%
2466 }

```

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

2467 \let\MT@curr@set@name\@empty

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

```

2468 \def\MT@declare@sets#1#2#3{%
2469   \KV@@sp@def\MT@curr@set@name{#2}%
2470   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
2471     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2472     \global\MT@let@nc{MT@#1list@size@\MT@curr@set@name}\@empty
2473   }%
2474   \global\MT@let@nc{MT@#1@set@@\MT@curr@set@name}\@empty
2475   <debug>\MT@info{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2476   \setkeys{MT@#1@set}{#3}%
2477 }
```

`\MT@define@set@key@`  $\langle\#1\rangle$  = font axis,  $\langle\#2\rangle$  = feature.

```

2478 \def\MT@define@set@key@#1#2{%
2479   \define@key{MT@#2@set}{#1}[]{}%
2480   \global\MT@let@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
2481   \MT@map@clist@n{##1}{%
2482     \KV@@sp@def\MT@val{###1}%
2483     \MT@get@highlevel{#1}%

```

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

```

2484     \MT@exp@two@n@g@addto@macro
2485     {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
2486     {\MT@val,}%
2487   }%

```

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

```

2488   \expandafter\g@addto@macro\expandafter\MT@font@sets
2489   \csname MT@#2list@#1@\MT@curr@set@name\endcsname
2490   <debug>\MT@info@n1{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
2491   }%
2492 }
```

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

```

2493 \def\MT@get@highlevel#1{%
2494   \expandafter\MT@test@ast\MT@val*\@nil\relax}%

```

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

```

2495   \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
2496   \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.

```

2497   }%
2498 }
```

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

```

2499 \def\MT@test@ast#1*#2\@nil{%
2500   \def\@tempa{#1}%
2501   \MT@ifempty{#2}%
2502 }
```

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

```

\MT@fix@font@set 2503 \let\MT@font@sets\@empty
2504 \def\MT@fix@font@set#1{%
2505   \xdef#1{#1}%
2506   \global\@onelevel@sanitize#1%
2507 }
```

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

```

2508 \def\MT@define@set@key@size#1{%
2509   \define@key{MT@#1@set}{size}[]{%
2510     \MT@map@clist@{##1}{%
2511       \KV@sp@def\MT@val{###1}%
2512       \expandafter\MT@get@range\MT@val--\@nil
2513       \ifx\MT@val\relax \else
2514         \expandafter\MT@xadd
2515         \csname MT@#1list@size@\MT@curr@set@name\endcsname
2516         {{{\MT@lower}{\MT@upper}\relax}}%
2517       \fi
2518     }%
2519   <debug>\MT@info@{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
2520   }%
2521 }
```

Font sizes may also be specified as ranges. This has been requested by Andreas Bhmann, who has also offered valuable help in implementing this. Now, it is for instance possible to set up different lists for fonts with optical sizes. (The MinionPro project is trying to do this for the OpenType version of Adobe's Minion. See <http://developer.berlios.de/projects/minionpro/>.)

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

```

\MT@lower 2522 \def\MT@get@range#1-#2-#3\@nil{%
2523   \MT@ifempty{#1}{%
2524     \MT@ifempty{#2}{%
2525       \let\MT@val\relax
2526     }%
2527     \def\MT@lower{0}%
2528     \def\MT@val{#2}%
2529     \MT@get@size
2530     \edef\MT@upper{\MT@val}%
2531   }%
2532 }{%
2533   \def\MT@val{#1}%
2534   \MT@get@size
2535   \ifx\MT@val\relax \else
2536     \edef\MT@lower{\MT@val}%
2537     \MT@ifempty{#2}{%
2538       \MT@ifempty{#3}%
2539       {\def\MT@upper{-1}}%

```

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

```

2540   {\def\MT@upper{2048}}%
2541 }{%
2542   \def\MT@val{#2}%
2543   \MT@get@size
2544   \ifx\MT@val\relax \else
2545     \MT@ifdim\MT@lower>\MT@val{%
2546       \MT@error{%
2547         Invalid size range (\MT@lower\space > \MT@val) in font set
2548         '~\MT@curr@set@name'.\MessageBreak Swapping sizes}}%
2549     \edef\MT@upper{\MT@lower}%
2550     \edef\MT@lower{\MT@val}%
2551   }{%
2552     \edef\MT@upper{\MT@val}%
2553   }%
2554   \MT@ifdim\MT@lower=\MT@upper
2555   {\def\MT@upper{-1}}%
2556   \relax

```

```

2557     \fi
2558     }%
2559     \fi
2560     }%
2561 }

```

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

```

2562 \def\MT@get@size{%
    A single star would mean \sizedefault, which doesn't exist, so we define it to be
    \normalsize.

```

```

2563     \if*\MT@val\relax
2564         \def\@tempa{\normalsize}%
2565     \else
2566         \MT@let@cn\@tempa{\MT@val}%
2567     \fi
2568     \ifx\@tempa\relax \else

```

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

```

2569     \begingroup
2570     \def\set@fontsize##1##2##3##4\@nil{\gdef\MT@val{##2}}%
2571     \@tempa\@nil
2572     \endgroup
2573     \fi

```

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

```

2574     \MT@ifdimen\MT@val{%
2575         \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2576         \edef\MT@val{\strip@pt\@tempdima}%
2577     }{%
2578         \MT@warning{Could not parse font size `~\MT@val'\MessageBreak
2579                     in font set `~\MT@curr@set@name'}%
2580         \let\MT@val\relax
2581     }%
2582 }

```

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

```

2583 \def\MT@define@set@key@font#1{%
2584     \define@key{MT@#1@set}{font}[]{}%
2585     \global\MT@let@nc{MT@#1list@font@MT@curr@set@name}\@empty
2586     \MT@map@clist@n{##1}{%
2587         \KV@esp@def\MT@val{###1}%
2588         \MT@ifstreq\MT@val*{\def\MT@val{*/**/*/*}}\relax
2589         \expandafter\MT@get@font\MT@val////\@nil
2590         \MT@exp@two@n@g@addto@macro
2591             {\csname MT@#1list@font@MT@curr@set@name\expandafter\endcsname}%
2592             {\MT@val,}%
2593     }%
2594     \expandafter\g@addto@macro\expandafter\MT@font@sets
2595         \csname MT@#1list@font@MT@curr@set@name\endcsname
2596     <debug>\MT@info@n{1}{-- font: \@nameuse{MT@#1list@font@MT@curr@set@name}}%
2597     }%
2598 }

```

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

```

2599 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%

```

```

2600 \MT@ifempty{#1#2#3#4#5}\relax{%
2601 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
2602 \ifx\MT@val\relax\def\MT@val{0}\fi
2603 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2604 \let\MT@val\@tempb
2605 }%
2606 }

```

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

```

2607 \def\MT@get@font@#1#2#3#4#5#6{%
2608 \let\@tempb\@empty
2609 \def\MT@temp{#1/#2/#3/#4/#5}%
2610 \MT@get@axis{encoding}{#1}%
2611 \MT@get@axis{family}{#2}%
2612 \MT@get@axis{series}{#3}%
2613 \MT@get@axis{shape}{#4}%
2614 \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
2615 \MT@ifempty{#5}{%
2616 \MT@warn@axis@empty{size}{\string\normalsize}%
2617 \def\MT@val{*}%
2618 }{%
2619 \def\MT@val{#5}%
2620 }%
2621 \MT@get@size
2622 }

```

`\MT@get@axis`

```

2623 \def\MT@get@axis#1#2{%
2624 \def\MT@val{#2}%
2625 \MT@get@highlevel{#1}%
2626 \MT@ifempty\MT@val{%
2627 \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2628 \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2629 }\relax
2630 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2631 }

```

`\MT@warn@axis@empty`

```

2632 \def\MT@warn@axis@empty#1#2{%
2633 \MT@warning{#1 axis is empty in font specification\MessageBreak
2634 \MT@temp'. Using '#2' instead}%
2635 }

```

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

```

2636 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
2637 \MT@define@set@key@{encoding}{#1}%
2638 \MT@define@set@key@{family}{#1}%
2639 \MT@define@set@key@{series}{#1}%
2640 \MT@define@set@key@{shape}{#1}%
2641 \MT@define@set@key@size{#1}%
2642 \MT@define@set@key@font{#1}%
2643 }

```

`\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.

```

2644 \renewcommand*\UseMicrotypeSet[2][{}]{%
2645 \KV@sp@def\@tempa{#1}%
2646 \MT@ifempty\@tempa{%
2647 \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
2648 }{%
2649 \MT@map@clist@c\@tempa{%

```

```

2650 \KV@sp@def\@tempa{##1}%
2651 \MT@ifempty\@tempa\relax{%
2652 \MT@is@feature{activation of set `#2'}{%
2653 \MT@exp@one@n\MT@use@set
2654 {\csname MT@rbba@\@tempa\endcsname}{#2}%
2655 }%
2656 }%
2657 }%
2658 }%
2659 }

```

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

```

\MT@ex@setname 2660 \def\MT@use@set#1#2{%
\MT@tr@setname 2661 \KV@sp@def\@tempa{#2}%
2662 \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@sp@setname 2663 \MT@xdef@n{MT@#1@setname}{\@tempa}%
\MT@kn@setname 2664 \MT@info{Using \@nameuse{MT@abbr@#1} set \@tempa'}%
\MT@use@set 2665 }%
2666 \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2667 \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2668 }%
2669 \MT@error{%
2670 The \@nameuse{MT@abbr@#1} set \@tempa' is undeclared.\MessageBreak
2671 Using set \@nameuse{MT@#1@setname}' instead}{}%
2672 }%
2673 }

```

\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.

```

2674 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2675 \KV@sp@def\@tempa{#1}%
2676 \MT@ifempty\@tempa{%
2677 \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
2678 }%
2679 \MT@map@clist@c\@tempa{%
2680 \KV@sp@def\@tempa{##1}%
2681 \MT@ifempty\@tempa\relax{%
2682 \MT@is@feature{declaration of default set `#2'}{%
2683 \MT@exp@one@n\MT@set@default@set
2684 {\csname MT@rbba@\@tempa\endcsname}{#2}%
2685 }%
2686 }%
2687 }%
2688 }%
2689 }

```

\MT@default@pr@set

```

\MT@default@ex@set 2690 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 2691 \KV@sp@def\@tempa{#2}%
2692 \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@default@sp@set 2693 <debug>\MT@info{1}{declaring default \@nameuse{MT@abbr@#1} set \@tempa'}%
\MT@default@kn@set 2694 \MT@xdef@n{MT@default@#1@set}{\@tempa}%
\MT@set@default@set 2695 }%
2696 \MT@error{%
2697 The \@nameuse{MT@abbr@#1} set \@tempa' is not declared.\MessageBreak
2698 Cannot make it the default set. Using set\MessageBreak `all' instead}{}%
2699 \MT@xdef@n{MT@default@#1@set}{all}%
2700 }%
2701 }

```



### 14.3.2 Variants and Aliases

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

```
2702 \let\MT@variants\@empty
2703 \def\DeclareMicrotypeVariants{%
2704   \ifstar
2705     \MT@DeclareVariants
2706   {\let\MT@variants\@empty\MT@DeclareVariants}%
2707 }
```

`\MT@DeclareVariants`

```
2708 \def\MT@DeclareVariants#1{%
2709   \MT@map@clist@n{#1}%
2710   \KV@sp@def\@tempa{##1}%
2711   \@onelevel@sanitize\@tempa
2712   \xdef\MT@variants{\MT@variants{\@tempa}}%
2713 }%
2714 }
```

`\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.

```
2715 \renewcommand*\DeclareMicrotypeAlias[2]{%
2716   \KV@sp@def\@tempa{#1}%
2717   \KV@sp@def\@tempb{#2}%
2718   \@onelevel@sanitize\@tempb
2719   \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2720     \MT@warning{Alias font family '\@tempb' will override
2721       alias \@nameuse{MT@\@tempa @alias}'\MessageBreak
2722       for font family '\@tempa'}}%
2723   \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.

```
2724   \MT@ifdefined@c@T{MT@family{%
2725     <debug>\MT@dinfo{1}{Activating alias font '\@tempb' for '\MT@family'}%
2726     \MT@gl@t{MT@familyalias\@tempb
2727   }%
2728 }
```

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

```
2729 \def\LoadMicrotypeFile#1{%
2730   \KV@sp@def\@tempa{#1}%
2731   \@onelevel@sanitize\@tempa
2732   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2733   \ifMT@inlist@
2734     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2735   \else
2736     \MT@xadd\MT@file@list{\@tempa,}%
2737     \MT@begin@catcodes
2738     \InputIfFileExists{mt-\@tempa.cfg}{%
2739       \edef\MT@curr@file{mt-\@tempa.cfg}%
2740       \MT@vinfo{... Loading configuration file \MT@curr@file}%
2741     }{%
2742       \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2743         does not exist}%
2744     }%
2745     \MT@end@catcodes
2746   \fi
2747 }
```

### 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@DisableLigatures`

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

```

\MT@nl@ligatures 2748 \MT@requires@pdftex5{
2749   \def\DisableLigatures{%
2750     \MT@begin@catcodes
2751     \MT@DisableLigatures
2752   }
2753   \newcommand*\MT@DisableLigatures[2][]{%
2754     \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
2755     \xdef\MT@active@features{\MT@active@features,nl}%
2756     \global\MT@noligaturestrue
2757     \MT@declare@sets{nl}{no ligatures}{#2}%
2758     \gdef\MT@nl@setname{no ligatures}%
2759     \MT@end@catcodes
2760   }
2761 }{
  If pdfTeX is too old, we issue a warning.
2762   \renewcommand*\DisableLigatures[2][]{%
2763     \MT@warning{Disabling ligatures of a font is only possible\MessageBreak
2764       with pdftex version 1.30 or newer.\MessageBreak
2765       Ignoring \string\DisableLigatures}%
2766   }
2767 }
```

### 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.

```

2768 \def\DeclareMicrotypeBabelHook#1#2{%
2769   \MT@map@clist@n{#1}{%
2770     \KV@esp@def\@tempa{##1}%
2771     \MT@gdef@n{MT@babel@\@tempa}{#2}%
2772   }%
2773 }
```

### 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).

```

2774 \def\SetProtrusion{%
2775   \MT@begin@catcodes
2776   \MT@SetProtrusion
2777 }
```

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

```

  \MT@pr@c@name 2778 \newcommand*\MT@SetProtrusion[3][]{%
\MT@extra@context 2779   \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.

```
2780 \MT@set@named@keys{MT@pr@c}{#1}%
2781 <debug>\MT@info{1}{creating protrusion list `~\MT@pr@c@name'}%
2782 \def\MT@permutelist{pr@c}%
2783 \setkeys{MT@cfg}{#2}%
```

We have parsed the second argument, and can now define macros for all permutations of the font attributes to point to \MT@pr@c@<name>, ...

```
2784 \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.

```
2785 \MT@gdef@n{MT@pr@c@MT@pr@c@name}{#3}%
2786 \MT@end@catcodes
2787 }
```

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

```
2788 \def\SetExpansion{%
2789 \MT@begin@catcodes
2790 \MT@SetExpansion
2791 }
```

\MT@SetExpansion

```
\MT@ex@c@name 2792 \newcommand*\MT@SetExpansion[3][]{%
\MT@extra@context 2793 \let\MT@extra@context\@empty
2794 \MT@set@named@keys{MT@ex@c}{#1}%
\MT@permutelist 2795 \MT@ifdefined@n{T}{MT@ex@c@MT@ex@c@name @factor}{%
2796 \ifnum\csize MT@ex@c@MT@ex@c@name @factor\endcsize > \@m
2797 \MT@warning@n{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
2798 too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
2799 maximum of 1000}%
2800 \global\MT@let@nc{MT@ex@c@MT@ex@c@name @factor}\@m
2801 \fi
2802 }%
2803 <debug>\MT@info{1}{creating expansion list `~\MT@ex@c@name'}%
2804 \def\MT@permutelist{ex@c}%
2805 \setkeys{MT@cfg}{#2}%
2806 \MT@permute
2807 \MT@gdef@n{MT@ex@c@MT@ex@c@name}{#3}%
2808 \MT@end@catcodes
2809 }
```

\SetTracking

```
2810 \def\SetTracking{%
2811 \MT@begin@catcodes
2812 \MT@SetTracking
2813 }
```

\MT@SetTracking Third argument may be empty.

```
2814 \newcommand*\MT@SetTracking[3][]{%
2815 \let\MT@extra@context\@empty
2816 \MT@set@named@keys{MT@tr@c}{#1}%
2817 <debug>\MT@info{1}{creating tracking list `~\MT@tr@c@name'}%
2818 \def\MT@permutelist{tr@c}%
2819 \setkeys{MT@cfg}{#2}%
2820 \MT@permute
2821 \KV@sp@def\@tempa{#3}%
2822 \MT@ifempty\@tempa\relax{%
2823 \MT@ifint\@tempa
```

```

2824      {\MT@xdef@n{MT@tr@cc@MT@tr@cc@name}{\@tempa}}%
2825      {\MT@warning{Value `~\@tempa' is not a number in\MessageBreak
2826               tracking set `~\MT@curr@set@name'}}}%
2827      \MT@end@catcodes
2828  }

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

\MT@SetExtraSpacing
2833  \newcommand*\MT@SetExtraSpacing[3] [] {%
2834      \let\MT@extra@context\@empty
2835      \MT@set@named@keys{MT@sp@c}{#1}%
2836      \MT@permutelist{#2}{creating spacing list `~\MT@sp@c@name'}%
2837      \def\MT@permutelist{sp@c}%
2838      \setkeys{MT@cfc}{#2}%
2839      \MT@permute
2840      \MT@gdef@n{MT@sp@c@MT@sp@c@name}{#3}%
2841      \MT@end@catcodes
2842  }

\SetExtraKerning
2843  \def\SetExtraKerning{%
2844      \MT@begin@catcodes
2845      \MT@SetExtraKerning
2846  }

\MT@SetExtraKerning
2847  \newcommand*\MT@SetExtraKerning[3] [] {%
2848      \let\MT@extra@context\@empty
2849      \MT@set@named@keys{MT@kn@c}{#1}%
2850      \MT@permutelist{#2}{creating kerning list `~\MT@kn@c@name'}%
2851      \def\MT@permutelist{kn@c}%
2852      \setkeys{MT@cfc}{#2}%
2853      \MT@permute
2854      \MT@gdef@n{MT@kn@c@MT@kn@c@name}{#3}%
2855      \MT@end@catcodes
2856  }

\MT@set@named@keys    We first set the name (if specified), then remove it from the list, and set the
\MT@options           remaining keys.
2857  \def\MT@set@named@keys#1#2{%
2858      \def\x##1name=##2,##3\@nil{%
2859          \setkeys{#1}{name=##2}%
2860          \def\MT@options{##1##3}%
2861          \MT@rem@from@clist{name=}\MT@options
2862      }%
2863      \x#2,name=,\@nil
2864      \@expandtwoargs\setkeys{#1}\MT@options
2865  }

\MT@define@code@key    Define the keys for the configuration lists (which are setting the codes, in pdfTeX
                        speak).
2866  \def\MT@define@code@key#1#2{%
2867      \define@key{MT@#2}{#1} [] {%
2868          \@tempcnta=\@ne
2869          \MT@map@clist@n{##1}%
2870          \KV@sp@def\MT@val{###1}%

```

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

```

2871 \MT@get@highlevel{#1}%
2872 \MT@edef\MT@temp#1\the\@tempcnta\MT@val}%
2873 \advance\@tempcnta \@ne
2874 }%
2875 }%
2876 }

```

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

```

2877 \def\MT@define@code@key@size#1{%
2878 \define@key{MT@#1}{size}[]{%
2879 \MT@map@clist@n{##1}%
2880 \KV@sp@def\MT@val{###1}%
2881 \expandafter\MT@get@range\MT@val--\@nil
2882 \ifx\MT@val\relax \else
2883 \expandafter\MT@xadd\csname MT@temp#size\endcsname
2884 {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2885 \fi
2886 }%
2887 }%
2888 }

```

\MT@define@code@key@font

```

2889 \def\MT@define@code@key@font#1{%
2890 \define@key{MT@#1}{font}[]{%
2891 \MT@map@clist@n{##1}%
2892 \KV@sp@def\MT@val{###1}%
2893 \MT@ifstreq\MT@val*\def\MT@val{*/*/*/*}\relax
2894 \expandafter\MT@get@font@and@size\MT@val////\@nil
2895 \MT@xdef\MT@MT@permutelist @\@tempb\MT@extra@context}%
2896 {\csname MT@MT@permutelist @name\endcsname}%
2897 <debug>\MT@info@n1{1}{initialising: use list for font \@tempb=\MT@val
2898 <debug> \ifx\MT@extra@context\@empty\else\MessageBreak
2899 <debug> (context: \MT@extra@context)\fi}%
2900 \expandafter\MT@xaddb
2901 \csname MT@MT@permutelist @\@tempb\MT@extra@context @size\endcsname
2902 {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
2903 }%
2904 }%
2905 }

```

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

```

2906 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2907 \MT@ifempty{#1#2#3#4#5}\relax
2908 {\MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}}%
2909 }
2910 \MT@define@code@key{encoding}{cfg}
2911 \MT@define@code@key{family}{cfg}
2912 \MT@define@code@key{series}{cfg}
2913 \MT@define@code@key{shape}{cfg}
2914 \MT@define@code@key@size{cfg}
2915 \MT@define@code@key@font{cfg}

```

\MT@define@opt@key

```

2916 \def\MT@define@opt@key#1#2{%
2917 \define@key{MT@#1c}{#2}[]{\MT@ifempty{##1}\relax%
2918 \MT@xdef\MT@#1c@\MT@curr@set@name @#2{##1}}}%
2919 }

```

The options in the optional first argument.

```

2920 \MT@map@clist@c\MT@features{%

```

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

```

2921 \define@key{MT@#1@c}{name}[] {%
2922   \MT@ifempty{##1} {%
2923     \MT@edefn{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2924   } {%
2925     \MT@edefn{MT@#1@c@name}{##1}%
2926     \MT@ifdefinedn@T{MT@#1@c@csname MT@#1@c@name\endcsname} {%
2927       \MT@warning{Redefining \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}}%
2928     }%
2929   }%
2930   \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
2931 }%
2932 \MT@define@opt@key{#1}{load}%
2933 \MT@define@opt@key{#1}{factor}%
2934 \MT@define@opt@key{#1}{preset}%
2935 \MT@define@opt@key{#1}{inputenc}%

```

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

```

2936 \define@key{MT@#1@c}{context}[] {\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
2937 }

```

Automatically enable font copying if we find an expansion context. After the preamble, check whether font copying is enabled. For older pdfTeX versions, disallow. Also disable for LuaTeX.

```

2938 \MT@requires@pdftex7{
2939   *lua
2940   \MT@requires@luatex{
2941     \define@key{MT@ex@c}{context}[] {%
2942       \MT@error{Expansion contexts currently don't work with luatex.\MessageBreak
2943         Ignoring `context' key\on@line}%
2944       {Use pdftex instead.}%
2945     }
2946   }{
2947     /lua
2948     \define@key{MT@ex@c}{context}[] {\MT@ifempty{#1}\relax{%
2949       \MT@gl@t\MT@copy@font\MT@copy@font@
2950       \def\MT@extra@context{#1}}}%
2951   }
2952   \MT@addto@setup{%
2953     \define@key{MT@ex@c}{context}[] {%
2954       \ifx\MT@copy@font\MT@copy@font@
2955         \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
2956       \else
2957         \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
2958           Ignoring `context' key\on@line}%
2959         {Either move the settings inside the preamble,\MessageBreak
2960           or load the package with the `copyfonts' option.}%
2961       \fi
2962     }
2963   }
2964   lua }
2965 }{
2966   \define@key{MT@ex@c}{context}[] {%
2967     \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
2968       or later. Ignoring `context' key\on@line}%
2969     {Upgrade pdftex.}%
2970   }
2971 }

```

\MT@warn@nodim

```

2972 \def\MT@warn@nodim#1{%
2973   \MT@warning{'\@tempa' is not a dimension.\MessageBreak
2974             Ignoring it and setting values relative to\MessageBreak #1}%
2975 }

```

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

```

2976 \define@key{MT@pr@c}{unit}[character]{%
2977   \global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
2978   \def\@tempa{#1}%
2979   \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.

```

2980     \MT@ifdimen\@tempa
2981     {\global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
2982     {\MT@warn@nodim{character widths}}%
2983   }%
2984 }

```

Tracking may only be relative to a dimension.

```

2985 \define@key{MT@tr@c}{unit}[1em]{%
2986   \global\MT@let@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
2987   \def\@tempa{#1}%
2988   \MT@ifdimen\@tempa
2989   {\global\MT@let@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
2990   {\MT@warn@nodim{1em}%
2991     \MT@gdefn{MT@tr@c@MT@curr@set@name @unit}{1em}}%
2992 }

```

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

```

2993 \MT@map@clist@n{sp,kn}{%
2994   \define@key{MT@#1@c}{unit}[space]{%
2995     \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
2996     \def\@tempa{##1}%
2997     \MT@ifstreq\@tempa{character}\relax{%
2998       \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
2999       \MT@ifstreq\@tempa{space}\relax{%
3000         \MT@ifdimen\@tempa
3001         {\global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
3002         {\MT@warn@nodim{width of space}}%
3003       }%
3004     }%
3005   }%
3006 }

```

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

```

3007 \MT@map@clist@n{stretch,shrink,step}{%
3008   \define@key{MT@ex@c}{#1}[]{%
3009     \MT@ifempty{##1}\relax{%
3010       \MT@ifint{##1}{%

```

A space terminates the number.

```

3011       \MT@gdefn{MT@ex@c@MT@curr@set@name @#1}{##1}%
3012     }%
3013     \MT@warning{%
3014       Value `##1' for option `#1' is not a number.\MessageBreak
3015       Ignoring it}%
3016   }%
3017 }%
3018 }%
3019 }
3020 \define@key{MT@ex@c}{auto}[true]{%

```

```

3021 \def\@tempa{#1}%
3022 \csname if\@tempa\endcsname

Don't use autoexpand for pdfTeX version older than 1.20.
3023 \MT@requires@pdfTeX4{%
3024   \MT@gdef\@tempa{\MT@curr@set@name @auto}{autoexpand}%
3025 }{%
3026   \MT@warning{pdfTeX too old for automatic font expansion}%
3027 }
3028 \else
3029   \MT@requires@pdfTeX4{%
3030     \global\MT@let\@tempa{\MT@curr@set@name @auto}\@empty
3031   }\relax
3032 \fi
3033 }

```

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

```

3034 \MT@define@opt@key{tr}{spacing}
3035 \MT@define@opt@key{tr}{outerspacing}
3036 \MT@define@opt@key{tr}{outerkerning}

Which ligatures should be disabled?
3037 \define@key{MT@tr@{c}}{noligatures}[]{}
3038 {\MT@xdef\@tempa{\MT@curr@set@name @noligatures}{#1}}
3039 \define@key{MT@tr@{c}}{outer spacing}[]{\setkeys{MT@tr@{c}}{outerspacing={#1}}}
3040 \define@key{MT@tr@{c}}{outer kerning}[]{\setkeys{MT@tr@{c}}{outerkerning={#1}}}
3041 \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.

```

3042 \renewcommand*\DeclareCharacterInheritance[1][]{%
3043   \let\MT@extra@context\@empty
3044   \let\MT@extra@inputenc\@undefined
3045   \let\MT@inh@feat\@empty
3046   \setkeys{MT@inh@}{#1}%
3047   \MT@begin@catcodes
3048   \MT@set@inh@list
3049 }

```

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

```

3050 \def\MT@set@inh@list#1#2{%
3051   \MT@ifempty\MT@inh@feat{%
3052     \MT@map@clist@{c}\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
3053   }{%
3054     \MT@map@clist@{c}\MT@inh@feat{{%
3055       \KV@{sp@def}\@tempa{##1}%
3056       \MT@ifempty\@tempa\relax%
3057       \MT@exp@one@n\MT@declare@char@inh
3058       {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%

```



```

3059     }%
3060   }}%
3061 }%
3062 \MT@end@catcodes
3063 }

```

The keys for the optional argument.

```

3064 \MT@map@clist@c\MT@features@long{%
3065   \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}
3066   \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}

```

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

```

3067 \def\MT@declare@char@inh#1#2#3{%
3068   \MT@edef@n{MT@#1@inh@name}%
3069   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3070   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3071   \MT@ifdefined@cT\MT@extra@inputenc{%
3072     \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3073   <debug>\MT@debug{1}{creating inheritance list "\@nameuse{MT@#1@inh@name}"}%
3074   \MT@gdef@n{MT@#1@inh@csname MT@#1@inh@name@endcsname}{#3}%
3075   \def\MT@permute@list{#1@inh}%
3076   \setkeys{MT@inh}{#2}%
3077   \MT@permute
3078 }

```

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

```

3079 \define@key{MT@inh}{encoding}[]{%
3080   \def\MT@val{#1}%
3081   \expandafter\MT@encoding@check\MT@val,\@nil
3082   \MT@get@highlevel{encoding}%
3083   \MT@edef@n{MT@tempencoding1}{\MT@val}%
3084 }

```

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

```

3085 \def\MT@encoding@check#1,#2\@nil{%
3086   \MT@ifempty{#2}\relax{%
3087     \edef\MT@val{#1}%
3088     \MT@warning{You may only specify one encoding for character\MessageBreak
3089               inheritance lists. Ignoring encoding(s) #2}%
3090   }%
3091 }

```

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

```

3092 \MT@define@code@key{family}{inh}
3093 \MT@define@code@key{series}{inh}
3094 \MT@define@code@key{shape}{inh}
3095 \MT@define@code@key@size{inh}
3096 \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).

```

3097 \def\MT@inh@do#1,{%
3098   \ifx\relax#1\empty \else
3099     \MT@inh@split #1==\relax
3100     \expandafter\MT@inh@do
3101   \fi
3102 }

```

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

```

3103 \def\MT@inh@split#1=#2=#3\relax{%
3104   \def\@tempa{#1}%
3105   \ifx\@tempa\@empty \else
3106     \MT@get@slot
3107     \ifnum\MT@char > \m@ne
3108       \let\MT@val\MT@char
3109       \MT@map@clist@n{#2}{%
3110         \def\@tempa{##1}%
3111         \ifx\@tempa\@empty \else
3112           \MT@get@slot
3113           \ifnum\MT@char > \m@ne
3114             \expandafter\MT@xadd
3115             \csname MT@inh@\MT@listname @\MT@val @\endcsname
3116             {{\MT@char}}}%
3117           \fi
3118         \fi
3119       }%
3120   \debug\MT@info@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 be the expansion of `\MT@{list type}@name`, i. e., the name of the currently defined list. `\MT@permute@@` Size ranges are held in a separate macro called `\MT@{list type}@/{font axes}@sizes`, which in turn contains the respective *list name*s attached to the ranges. `\MT@permute@@@`

```

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@gl@et\MT@temp@size\@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@temp@encoding\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@temp@family\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@temp@series\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 \def\@tempb{////}%
3178 \ifx\@tempa\@tempb \else
3179 \ifx\MT@tempencoding\@empty
3180 \MT@warning{%
3181 You have to specify an encoding for\MessageBreak
3182 \nameuse{MT@abbr@MT@permutelist} list
3183 ~\nameuse{MT@MT@permutelist @name}'.\MessageBreak
3184 Ignoring it}%
3185 \else
3186 \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

3187 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3188 \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3189 }%
3190 \expandafter\MT@xaddb
3191 \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
3192 \MT@tempsize
3193 <debug>\MT@info@n1{1}{initialising: use list for font \@tempa,\MessageBreak
3194 <debug> sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3195 <debug> @sizes\endcsname}%
3196 }{%

```

Only one list can apply to a given combination.

```

3197 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3198 \MT@warning{\nameuse{MT@abbr@MT@permutelist} list
3199 ~\nameuse{MT@MT@permutelist @name}' will override list\MessageBreak

```

```

3200         \@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3201         for font \@tempa'%
3202     }%
3203 <debug>\MT@edinfo@n1{1}{initialising: use list for font \@tempa
3204 <debug>         \ifx\MT@extra@context\@empty\else\MessageBreak
3205 <debug>         (context: \MT@extra@context)\fi}%
3206     }%
3207     \MT@xdefn{MT@MT@permutelist @\@tempa\MT@extra@context}%
3208     {\csname MT@MT@permutelist @name\endcsname}%
3209     \fi
3210 \fi
3211 }

```

\MT@permute@define     Define the commands.

```

3212 \def\MT@permute@define#1{%
3213   \expandafter\@tempcnta=\csname MT@cnt@#1\endcsname\relax
3214   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3215   {\MT@edefn{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3216   {\MT@let@nc{MT@temp#1}\@empty}%
3217 }

```

\MT@permute@reset     Reset the commands.

```

3218 \def\MT@permute@reset#1{%
3219   \@tempcnta=\@ne
3220   \MT@loop
3221   \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3222   \advance\@tempcnta\@ne
3223   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3224   \iftrue
3225   \iffalse
3226   \MT@repeat
3227 }

```

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

```

3228 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

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

```

3229 \def\MT@check@rlist@#1#2#3{%
3230   \def\@tempb{#1}%
3231   \def\@tempc{#2}%
3232   \MT@if@false
3233   \expandafter\MT@map@tlist@c
3234   \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
3235   \MT@check@range
3236 }

```

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

```

3237 \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.

```

3238 \def\MT@check@range@#1#2#3{%
3239   \MT@ifdim{#2}=\m@ne{%
3240     \MT@ifdim\@tempc=\m@ne{%

```

- Both items are simple sizes.

```

3241     \MT@ifdim\@tempb={#1}\MT@if@true\relax
3242   }{%

```

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

```

3243 \MT@ifdim\@tempb>{#1}\relax{%
3244 \MT@ifdim\@tempc>{#1}{%
3245 \MT@iftrue
3246 \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3247 }\relax
3248 }%
3249 }%
3250 }{%
3251 \MT@ifdim\@tempc=\m@ne{%

```

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

```

3252 \MT@ifdim\@tempb<{#2}{%
3253 \MT@ifdim\@tempb<{#1}\relax\MT@iftrue
3254 }\relax
3255 }{%

```

- Both items are ranges.

```

3256 \MT@ifdim\@tempb<{#2}{%
3257 \MT@ifdim\@tempc>{#1}{%
3258 \MT@iftrue
3259 \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3260 }\relax
3261 }\relax
3262 }%
3263 }%
3264 \ifMT@if@
3265 \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3266 \@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
3267 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.

```

3268 \expandafter\MT@tlist@break
3269 \fi
3270 }

```

## 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 3271 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 3272 \newif\ifMT@opt@auto
3273 \newif\ifMT@opt@DVI

```

`\MT@optwarn@admissible` Some warnings.

```

3274 \def\MT@optwarn@admissible#1#2{%
3275 \MT@warning@n1{`#1' is not an admissible value for option\MessageBreak
3276 `#2'. Assuming `false'}%
3277 }

```

`\MT@optwarn@nan`

```

3278 </package>
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 <*/package>

```

\MT@opt@def@set

```

3284 \def\MT@opt@def@set#1{%
3285   \MT@ifdefined@n@TF{MT@ \@tempb @set@@\MT@val}{%
3286     \MT@xdef@n{MT@ \@tempb @setname}{\MT@val}%
3287   }{%
3288     \MT@xdef@n{MT@ \@tempb @setname}{\@nameuse{MT@default@ \@tempb @set}}%
3289     \MT@warning@n{The #1 set `~\MT@val' is undeclared.\MessageBreak
3290       Using set ~\@nameuse{MT@ \@tempb @setname}' instead}%
3291   }%
3292 }
```

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

```

3293 \MT@map@clist@n{protrusion,expansion}{%
3294   \define@key{MT}{#1}[true]{%
3295     \csname MT@opt@#1true\endcsname
3296     \MT@map@clist@n{##1}{%
3297       \KV@sp@def\MT@val{###1}%
3298       \MT@ifempty\MT@val\relax{%
3299         \csname MT@#1true\endcsname
3300         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3301         \MT@ifstreq\MT@val{true}\relax
3302         {%
3303           \MT@ifstreq\MT@val{false}{%
3304             \csname MT@#1false\endcsname
3305           }{%
3306             \MT@ifstreq\MT@val{compatibility}{%
3307               \MT@let@nc{MT@ \@tempb @level}\@ne
3308             }{%
3309               \MT@ifstreq\MT@val{nocompatibility}{%
3310                 \MT@let@nc{MT@ \@tempb @level}\tw@
3311               }{%

```

If everything failed, it should be a set name.

```

3312         \MT@opt@def@set{#1}%
3313       }%
3314     }%
3315   }%
3316 }%
3317 }%
3318 }%
3319 }%
3320 }
```

activate is a shortcut for protrusion and expansion.

```

3321 \define@key{MT}{activate}[true]{%
3322   \setkeys{MT}{protrusion=#1}%
3323   \setkeys{MT}{expansion=#1}%
3324 }
```

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

```

3325 \MT@map@clist@n{spacing,kerning,tracking}{%
3326   \define@key{MT}{#1}[true]{%
3327     \MT@map@clist@n{##1}{%
3328       \KV@sp@def\MT@val{###1}%
3329       \MT@ifempty\MT@val\relax{%
3330         \csname MT@#1true\endcsname
3331         \MT@ifstreq\MT@val{true}\relax
3332         {%
3333           \MT@ifstreq\MT@val{false}{%
3334             \csname MT@#1false\endcsname
3335           }{%

```

```

3336         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3337         \MT@opt@def@set{#1}%
3338     }%
3339 }%
3340 }%
3341 }%
3342 }%
3343 }

```

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

```

3344 \def\MT@def@bool@opt#1#2{%
3345     \define@key{MT}{#1}[true]{%
3346         \def\@tempa{##1}%
3347         \MT@ifstreq\@tempa{true}\relax{%
3348             \MT@ifstreq\@tempa{false}\relax{%
3349                 \MT@optwarn@admissible{##1}{#1}%
3350                 \def\@tempa{false}%
3351             }%
3352         }%
3353         #2%
3354     }%
3355 }

```

Boolean options that only set the switch.

```

3356 \MT@map@clist@n{draft,selected,babel}{%
3357     \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
3358 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotruer}

```

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

```

3359 \MT@def@bool@opt{DVIoutput}{%
3360     \csname if\@tempa\endcsname
3361     \ifnum\pdfoutput>\z@ \MT@opt@DVIttrue \fi
3362     \pdfoutput\z@
3363 \else
3364     \ifnum\pdfoutput<\@ne \MT@opt@DVIttrue \fi
3365     \pdfoutput\@ne
3366 \fi
3367 }

```

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.

```

3368 \MT@def@bool@opt{defersetup}{%
3369     \csname if\@tempa\endcsname \else
3370         \AtEndOfPackage{%
3371             \MT@setup@
3372             \let\MT@setup@\@empty
3373             \let\MT@addto@setup@\firstofone
3374         }%
3375     \fi
3376 }

```

copyfonts will copy all fonts before setting them up. This allows expansion with different parameters. This options is also *undocumented* in the hope that we can

always find out automatically whether it's required.

```

3377 \MT@requires@pdftex7{
3378   \lua
3379   \MT@requires@luatex{
3380     \MT@def@bool@opt{copyfonts}{%
3381       \csname if\@tempa\endcsname
3382       \MT@error{The `copyfonts' option doesn't work with luatex}
3383       {Use pdftex instead.}%
3384     \fi
3385   }
3386 }{
3387   \lua
3388   \MT@def@bool@opt{copyfonts}{%
3389     \csname if\@tempa\endcsname
3390     \MT@gllet\MT@copy@font\MT@copy@font@
3391     \else
3392     \MT@gllet\MT@copy@font\relax
3393     \fi
3394   }
3395   \lua }
3396 }{
3397   \MT@def@bool@opt{copyfonts}{%
3398     \csname if\@tempa\endcsname
3399     \MT@error{The pdftex version you are using is too old\MessageBreak
3400       to use the `copyfonts' option}
3401     {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 \MT@map@clist@n{%
3432 <package> stretch,shrink,step,%
3433 letterspace}{%
3434 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
3435 \def\@tempa{##1 }%

```

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

```

3436 \MT@ifint\@tempa
3437 {\MT@edef@n{MT@#1}{\@tempa}}%
3438 {\MT@optwarn@nan{##1}{#1}}%
3439 }%
3440 }
3441 <*package>

```

factor will define the protrusion factor only.

```

3442 \define@key{MT}{factor}[\MT@factor@default]{%
3443 \def\@tempa{#1 }%
3444 \MT@ifint\@tempa
3445 {\edef\MT@pr@factor{\@tempa}}
3446 {\MT@optwarn@nan{#1}{factor}}%
3447 }

```

Unit for protrusion codes.

```

3448 \define@key{MT}{unit}{character}{%
3449 \def\@tempa{#1}%
3450 \MT@ifstreq\@tempa{character}\relax{%
3451 \MT@ifdimen\@tempa
3452 {\let\MT@pr@unit\@tempa}%
3453 {\MT@warning@n1{\@tempa' is not a dimension.\MessageBreak
3454 Ignoring it and setting values relative to\MessageBreak
3455 character widths}}%
3456 }%
3457 }

```

#### 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 T<sub>E</sub>X systems are switching to the pdfT<sub>E</sub>X engine even for DVI output, so that the user might not even be aware of the fact that she's running pdfT<sub>E</sub>X.)

```

3458 \MT@protrusiontrue
3459 \ifnum\pdfoutput<\@ne \else

```

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

```

3460 \MT@requires@pdftex4{
3461 \MT@expansiontrue
3462 \MT@autottrue
3463 }\relax
3464 \fi

```

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

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

```

3465 \define@key{MT}{config}[]{\relax}
3466 \def\MT@get@config#1config=#2,#3\@nil{%

```

```

3467 \MT@ifempty{#2}%
3468   {\def\MT@config@file{\MT@MT.cfg}}%
3469   {\def\MT@config@file{#2.cfg}}%
3470 }
3471 \expandafter\expandafter\expandafter\MT@get@config
3472 \csname opt@\currname.\@currxt\endcsname,config=,\@nil

```

Load the file.

```

3473 \IfFileExists{\MT@config@file}{%
3474   \MT@info@nl{Loading configuration file \MT@config@file}%
3475   \MT@begin@catcodes
3476   \let\MT@begin@catcodes\relax
3477   \let\MT@end@catcodes\relax
3478   \let\MT@curr@file\MT@config@file
3479   \input{\MT@config@file}%
3480   \endgroup
3481 }{\MT@warning@nl{%
3482   Could not find configuration file \MT@config@file!\MessageBreak
3483   This will almost certainly cause undesired results.\MessageBreak
3484   Please fix your installation}%
3485 }

```

If no default font set has been declared in the main configuration file, we use the (empty, possibly non-existent) ‘all’ set.

```

3486 \MT@map@clic@c\MT@features{%
3487   \MT@ifdefined@n@TF{MT@default@#1@set}\relax
3488   {\MT@gdef@n{MT@default@#1@set}{all}}%
3489 }

```

`\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).

```

3490 \def\MT@check@active@set#1{%
3491   \MT@ifdefined@n@TF{MT@#1@setname}{%
3492     \MT@info@nl{Using \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}}%
3493   }{%
3494     \global\MT@let@nn{MT@#1@setname}{MT@default@#1@set}%
3495     \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}}%
3496   }%
3497 }

```

### 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.

```

3498 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
3499   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3500   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
3501 \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 (un-  
\MT@define@optionX   less defersetup=false). In the document body, it accepts the options: protrusion,  
expansion, activate, tracking, spacing and kerning. Specifying font sets is not  
allowed.

```

3502 \def\microtypesetup{\setkeys{MT}}
3503 \MT@addto@setup{\def\microtypesetup{\setkeys{MTX}}}
3504 \def\MT@define@optionX#1#2{%
3505   \define@key{MTX}{#1}[true]{%
3506     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3507     \MT@map@clist@n{##1}{%
3508       \KV@sp@def\MT@val{###1}%
3509       \MT@ifempty\MT@val\relax{%
3510         \@tempcnta=\m@ne
3511       \MT@ifstreql\MT@val{true}{%

```

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

```

3512   \MT@checksetup{#1}{%
3513     \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
3514     \MT@info{Enabling #1
3515       (level \number\csname MT@\@tempb @level\endcsname)}%
3516   }%
3517 }{%
3518   \MT@ifstreql\MT@val{false}{%
3519     \@tempcnta=\z@
3520     \MT@info{Disabling #1}%
3521   }{%
3522     \MT@ifstreql\MT@val{compatibility}{%
3523       \MT@checksetup{#1}{%
3524         \@tempcnta=\@ne
3525         \MT@let@nc{MT@\@tempb @level}\@ne
3526         \MT@info{Setting #1 to level 1}%
3527       }%
3528     }{%
3529       \MT@ifstreql\MT@val{nocompatibility}{%
3530         \MT@checksetup{#1}{%
3531           \@tempcnta=\tw@
3532           \MT@let@nc{MT@\@tempb @level}\tw@
3533           \MT@info{Setting #1 to level 2}%
3534         }%

```

```

3535         }{\MT@warning{%
3536             Value `~\MT@val' for key `~#1' not recognised.\MessageBreak
3537             Use any of `true', `false', `compatibility' or~\MessageBreak
3538             `nocompatibility'}%
3539         }%
3540     }%
3541 }%
3542 }%
3543 \ifnum~\@tempcnta>\m@ne
3544 #2~\@tempcnta~\relax
3545 \fi
3546 }%
3547 }%
3548 }%
3549 }

```

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

```

3550 \def~\MT@checksetup#1{%
3551     \expandafter~\csname ifMT@#1~\endcsname
3552     \expandafter~\@firstofone
3553     \else
3554         \MT@warning{You cannot enable #1 if it was disabled~\MessageBreak
3555             in the package options,}%
3556         \expandafter~\@gobble
3557     \fi
3558 }

3559 \MT@define@optionX{protrusion}{pdfprotrudechars}
3560 \MT@define@optionX{expansion}{pdfadjustspacing}

```

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

```

3561 \MT@requires~\pdfTeX6{
3562     \def~\MT@define@optionX@#1#2{%
3563         \define@key{MTX}{#1}[true]{%
3564             \MT@map@clist@n{##1}{%
3565                 \KV@sp~\def~\MT@val{###1}%
3566                 \MT@ifempty~\MT@val~\relax{%
3567                     \@tempcnta=\m@ne
3568                     \MT@ifstreql~\MT@val{true}{%
3569                         \MT@checksetup{#1}%
3570                         \@tempcnta=\@ne
3571                         \MT@info{Enabling #1}%
3572                     }%
3573                 }{%
3574                     \MT@ifstreql~\MT@val{false}{%
3575                         \@tempcnta=\z@
3576                         \MT@info{Disabling #1}%
3577                     }{\MT@warning{Value `~\MT@val' for key `~#1' not recognised.
3578                         \MessageBreak Use either `true' or `false'}%
3579                     }%
3580                 }%
3581                 \ifnum~\@tempcnta>\m@ne
3582                     #2~\relax
3583                 \fi
3584             }%
3585         }%
3586     }%
3587 }

3588 \MT@define@optionX@{tracking}{\ifnum~\@tempcnta=\z@ \let~\MT@tracking~\relax~\else
3589     \let~\MT@tracking~\MT@tracking@~\fi}
3590 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue~\@tempcnta}

```

```

3591 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3592 \pdfappendkern \@tempcnta}
3593 }{

```

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

```

3594 \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
3595 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
3596 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
3597 }
3598 \define@key{MTX}{activate}[true]{%
3599 \setkeys{MTX}{protrusion={#1}}%
3600 \setkeys{MTX}{expansion={#1}}%
3601 }

```

\MT@saved@setupfont Disable everything – may be used as a work-around in case setting up fonts doesn't work in certain environments. (*Undocumented.*)

```

3602 \let\MT@saved@setupfont\MT@setupfont
3603 \define@key{MTX}{disable}[]{%
3604 \MT@info{Inactivate '\MT@MT' package}%
3605 \let\MT@setupfont\relax
3606 }
3607 \define@key{MTX}{enable}[]{%
3608 \MT@info{Reactivate '\MT@MT' package}%
3609 \let\MT@setupfont\MT@saved@setupfont
3610 }
3611 </package>

```

#### 14.4.5 Processing the Options

\MT@ProcessOptionsWithKV Parse options.

```

3612 <plain> \ifnum\MT@plain<2
3613 \def\MT@ProcessOptionsWithKV#1{%
3614 \let\@tempc\relax
3615 \let\MT@temp\@empty
3616 <plain> \ifnum\MT@plain=0
3617 \MT@map@clist@c\@classoptionslist{%
3618 \def\CurrentOption{##1}%
3619 \MT@ifdefined@nT{KV@#1\CurrentOption}{%
3620 \edef\MT@temp{\MT@temp,\CurrentOption,%
3621 \@expandtwoargs\@removeelement\CurrentOption
3622 \@unusedoptionlist\@unusedoptionlist
3623 }%
3624 }%
3625 \edef\MT@temp{\noexpand\setkeys{#1}%
3626 {\MT@temp\@optionlist{\@currname.\@current}}}%

```

*plain* can also handle options.

```

3627 <*plain>
3628 \else
3629 \edef\MT@temp{\noexpand\setkeys{#1}%
3630 {\csname usepkg@options@usepkg@pkg@endcsname}}%
3631 \fi
3632 </plain>
3633 \MT@temp
3634 \MT@clear@options
3635 }

3636 \MT@ProcessOptionsWithKV{MT}
3637 <plain> \fi
3638 <*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).

```
3639 \MT@addto@setup{%
3640 \ifMT@draft
```

We disable most of what we've just defined in the 3640 lines above if we are running in draft mode.

```
3641 \MT@warning@n{\draft' option active.\MessageBreak
3642             Disabling all micro-typographic extensions.\MessageBreak
3643             This might lead to different line and page breaks}
3644 \let\MT@setupfont\relax
3645 \renewcommand*\LoadMicrotypeFile[1]{}
3646 \renewcommand*\microtypesetup[1]{}
3647 \renewcommand*\microtypecontext[1]{}
3648 \renewcommand*\lsstyle{}
3649 \else
```

For DVI output, the user must have explicitly passed the expansion option to the package.

```
3650 \ifnum\pdfoutput<\@ne
3651 \ifMT@opt@expansion \else
3652 \MT@expansionfalse
3653 \fi
3654 \fi
```

pdf<sub>TeX</sub> 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!

```
3655 \MT@info@n{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3656             \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
```

Fix the font sets.

```
3657 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
```

Protrusion.

```
3658 \ifMT@protrusion
3659 \edef\MT@active@features{\MT@active@features,pr}
3660 \pdfprotrudechars\MT@pr@level
3661 \MT@info@n{Character protrusion enabled (level \number\MT@pr@level)%
3662 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3663 factor: \number\MT@pr@factor\fi
3664 \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}
3665 \MT@check@active@set{pr}
3666 \else
3667 \let\MT@protrusion\relax
3668 \MT@info@n{No character protrusion}
3669 \fi
```

Expansion.

```
3670 \ifMT@expansion
```

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

```
3671 \ifnum\MT@stretch=\m@ne
```

```

3672     \let\MT@stretch\MT@stretch@default
3673     \fi

```

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

```

3674     \ifnum\MT@shrink=\m@ne
3675         \ifnum\MT@stretch>\z@
3676             \let\MT@shrink\MT@stretch
3677         \else
3678             \let\MT@shrink\MT@shrink@default
3679         \fi
3680     \fi

```

If step has not been specified, we will set it to  $\min(\text{stretch}, \text{shrink})/5$ , rounded off, minimum value 1.

```

3681     \ifnum\MT@step=\m@ne
3682         \ifnum\MT@stretch>\MT@shrink
3683             \ifnum\MT@shrink=\z@
3684                 \@tempcnta=\MT@stretch
3685             \else
3686                 \@tempcnta=\MT@shrink
3687             \fi
3688         \else
3689             \ifnum\MT@stretch=\z@
3690                 \@tempcnta=\MT@shrink
3691             \else
3692                 \@tempcnta=\MT@stretch
3693             \fi
3694         \fi
3695         \divide\@tempcnta 5\relax
3696     \else
3697         \@tempcnta=\MT@step
3698         \ifnum\@tempcnta=\z@
3699             \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
3700                 Setting it to one}
3701         \fi
3702     \fi
3703     \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3704     \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>A</sub>T<sub>E</sub>X* programme really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdf<sub>T</sub><sub>E</sub>X).

```

3705     \let\MT@auto\empty
3706     \ifMT@auto
3707         \MT@requires@pdftex4{%

```

We turn off automatic expansion if output mode is DVI.

```

3708         \ifnum\pdfoutput<\@ne
3709             \ifMT@opt@auto
3710                 \MT@warning@nl{%
3711                     Automatic font expansion only works for PDF output.\MessageBreak
3712                     However, you are creating a DVI file. I will switch\MessageBreak
3713                     automatic font expansion off and hope that expanded\MessageBreak
3714                     fonts are available}
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@warning@nl{%
3722         The pdftex version you are using is too old for\MessageBreak
3723         automatic font expansion. I will switch it off and\MessageBreak
3724         hope that expanded fonts are available.\MessageBreak
3725         Otherwise, install pdftex version 1.20 or newer}
3726     \MT@autofalse
3727     \def\MT@auto{1000 }
3728 }
3729 \else

```

No automatic expansion.

```

3730     \MT@requires@pdftex4\relax{
3731         \def\MT@auto{1000 }
3732     }
3733 \fi

```

Choose the appropriate macro for selected expansion.

```

3734     \ifMT@selected
3735         \let\MT@set@ex@codes\MT@set@ex@codes@s
3736     \else
3737         \let\MT@set@ex@codes\MT@set@ex@codes@n
3738     \fi

```

Filter out stretch=0, shrink=0, since it would result in a pdf<sub>T</sub><sub>E</sub>X error.

```

3739     \ifnum\MT@stretch=\z@
3740         \ifnum\MT@shrink=\z@
3741             \MT@warning@nl{%
3742                 Both the stretch and shrink limit are set to zero.\MessageBreak
3743                 Disabling font expansion}
3744             \MT@expansionfalse
3745         \fi
3746     \fi
3747 \fi
3748 \ifMT@expansion
3749     \edef\MT@active@features{\MT@active@features,ex}%
3750     \pdfadjustspacing\MT@ex@level
3751     \MT@info@nl{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
3752         (level \number\MT@ex@level),\MessageBreak
3753         stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3754         step: \number\MT@step, \ifMT@selected\else non-\fi selected%
3755         \ifx\MT@copy@font\MT@copy@font@ .\MessageBreak
3756         Set up for expansion contexts (using font copies)\fi}

```

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

```

3757     \def\MT@check@step#1{%
3758         \@tempcnta=\csname MT@#1\endcsname
3759         \divide\@tempcnta \MT@step
3760         \multiply\@tempcnta \MT@step
3761         \ifnum\@tempcnta=\csname MT@#1\endcsname\else
3762             \MT@warning@nl{The #1 amount is not a multiple of step.\MessageBreak
3763                 The effective maximum #1 is \the\@tempcnta\space
3764                 (step \number\MT@step)}
3765         \fi
3766     }
3767     \MT@check@step{stretch}
3768     \MT@check@step{shrink}
3769     \MT@check@active@set{ex}

```

Inside \showhyphens, font expansion should be disabled.

```

3770     \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3771         \color@begingroup\everypar{}\parfillskip\z@skip
3772         \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne

```



```

3773      \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
\showhyphens  I wonder why it's defined globally (in ltfsbas.dtx)?
3774      \gdef\showhyphens#1{\setbox0\vbox{%
3775      \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3776      \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3777      \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
3778  \else
3779      \let\MT@expansion\relax
3780      \MT@info@n1{No font expansion}
3781  \fi
3782 }
3783 \MT@requires@pdftex6{
3784 </package>
3785 \MT@addto@setup{%
3786 <*package>

```

Tracking, spacing and kerning.

```

3787 \ifMT@tracking
3788 <*lua>
3789 \MT@requires@luatex{
3790 \MT@warning@n1{Tracking doesn't currently work with luatex}
3791 \let\MT@tracking\relax
3792 }{
3793 </lua>
3794 \edef\MT@active@features{\MT@active@features,tr}
3795 \MT@info@n1{Tracking enabled}
3796 \MT@check@active@set{tr}

```

Enable protrusion for compensation at the line edges.

```

3797 \ifMT@protrusion\else\pdfprotrudechars\@ne\fi
3798 <lua>
3799 \else
3800 \let\MT@tracking\relax
3801 \MT@info@n1{No tracking}
3802 \fi
3803 \ifMT@spacing
3804 \edef\MT@active@features{\MT@active@features,sp}
3805 \pdfadjustinterwordglue\@ne
3806 \MT@info@n1{Adjustment of interword spacing enabled}
3807 \MT@check@active@set{sp}
3808 \else
3809 \let\MT@spacing\relax
3810 \MT@info@n1{No adjustment of interword spacing}
3811 \fi
3812 \ifMT@kerning
3813 \edef\MT@active@features{\MT@active@features,kn}
3814 \pdfprependkern\@ne
3815 \pdfappendkern\@ne
3816 \MT@info@n1{Adjustment of character kerning enabled}
3817 \MT@check@active@set{kn}
3818 \else
3819 \let\MT@kerning\relax
3820 \MT@info@n1{No adjustment of character kerning}
3821 \fi
3822 </package>

```

\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.

```

3823 \ifnum\pdfoutput<\@ne

```

```

3824 \def\MT@warn@tracking@DVI{%
3825 \MT@warning@nl{%
3826 You are using tracking/letterspacing in DVI mode.\MessageBreak
3827 This will probably not work, unless the post-\MessageBreak
3828 processing program (dvips, dvipdfm(x), ...) is\MessageBreak
3829 able to create the virtual fonts on the fly}%
3830 \MT@glet\MT@warn@tracking@DVI\relax
3831 }
3832 \else
3833 \def\MT@warn@tracking@DVI{%
3834 \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
3835 \MT@glet\MT@warn@tracking@DVI\relax
3836 }
3837 \fi

3838 \ifnum\MT@letterspace=\m@ne
3839 \let\MT@letterspace\MT@letterspace@default
3840 \else
3841 \MT@ls@too@large\MT@letterspace
3842 \fi
3843 }

```

If pdfTeX is too old, we disable tracking, spacing and kerning.

```

3844 <*package>
3845 {}
3846 \MT@addto@setup{%
3847 \ifMT@tracking
3848 \MT@warning@nl{Tracking only works with pdftex version 1.40\MessageBreak
3849 or newer. Switching it off}%
3850 \else
3851 \MT@info@nl{No tracking (pdftex too old)}
3852 \fi
3853 \ifMT@spacing
3854 \MT@warning@nl{Adjustment of interword spacing only works with\MessageBreak
3855 pdftex version 1.40 or newer. Switching it off}%
3856 \else
3857 \MT@info@nl{No adjustment of interword spacing (pdftex too old)}
3858 \fi
3859 \ifMT@kerning
3860 \MT@warning@nl{Character kerning only works with\MessageBreak
3861 pdftex version 1.40 or newer. Switching it off}%
3862 \else
3863 \MT@info@nl{No adjustment of character kerning (pdftex too old)}
3864 \fi
3865 }
3866 }

```

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.

```

3867 \MT@requires@pdftex6{
3868 \AtBeginDocument{%
3869 \ifMT@spacing
3870 \ifMT@babel \else
3871 \ifnum\sfcode`. > 1500
3872 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
3873 \MT@warning@nl{%
3874 \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3875 interword spacing will disable it. You might want\MessageBreak
3876 to add ~\@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
3877 to your preamble}%

```

```

3878     }%
3879     \fi
3880   \fi
3881   \fi
3882 }
3883 }\relax

```

\DisableLigatures is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```

3884 \MT@requirespdfTeX5{
3885   \MT@addto@setup{%
3886     \ifMT@noligatures \else
3887       \let\MT@noligatures\relax
3888     \fi
3889   }
3890 }\relax

```

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

```

3891 \MT@addto@setup{%
3892   \ifx\MT@active@features\empty \else
3893     \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3894   \fi
3895   \MT@documenttrue
3896 }

```

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

```

3897 \def\MT@set@babel@context#1{%
3898   \MT@ifdefined@n@TF{MT@babel@#1}{%
3899     \MT@info{*** Changing to language context `#1'\MessageBreak\on@line}%
3900     \expandafter\MT@exp@one@n\expandafter\microtypecontext
3901     \csname MT@babel@#1\endcsname
3902   }{%
3903     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3904   }%
3905 }

```

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

```

3906 \ifpackageloaded{babel}{
3907   \def\MT@shorthandoff#1#2{%
3908     \MT@info@n1{Switching off #1 babel's active characters (#2)}%
3909     \shorthandoff{#2}}
3910 }{
3911   \def\MT@shorthandoff#1#2{%
3912     \MT@error{You must load the `babel' package before \MT@MT}
3913     {Otherwise, `~\MT@MT' cannot switch off the active characters
3914      for #1 babel.}}
3915 }

```

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

```

3916 \MT@addto@setup{%
3917   \ifMT@babel
3918     \ifpackageloaded{babel}{%
3919       \MT@info@n1{Redefining babel's language switching commands}
3920       \let\MT@orig@select@language\select@language
3921       \def\select@language#1{%
3922         \MT@orig@select@language{#1}%
3923         \MT@set@babel@context{#1}%
3924       }
3925       \let\MT@orig@foreign@language\foreign@language
3926       \def\foreign@language#1{%
3927         \MT@orig@foreign@language{#1}%

```

```

3928      \MT@set@babel@context{#1}%
3929      }
3930      \ifMT@kerning

```

Disable French babel's active characters.

```

3931      \MT@if@false
3932      \MT@with@babel@and@T{french} \MT@if@true
3933      \MT@with@babel@and@T{frenchb} \MT@if@true
3934      \MT@with@babel@and@T{francais} \MT@if@true
3935      \MT@with@babel@and@T{acadian} \MT@if@true
3936      \MT@with@babel@and@T{canadien} \MT@if@true
3937      \ifMT@if@MT@shorthandoff{French}{:;!}\fi

```

Disable Turkish babel's active characters.

```

3938      \MT@if@false
3939      \MT@with@babel@and@T{turkish} \MT@if@true
3940      \ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi
3941      \fi

```

In case babel was loaded before microtype:

```

3942      \MT@set@babel@context\language name
3943      }{%
3944      \MT@warning@n{You did not load the babel package.\MessageBreak
3945      The `babel' option won't have any effect}
3946      }
3947      \fi
3948      }

```

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

```

3949 \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.

```

3950 \AtBeginDocument\selectfont

```

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

```

3951 \edef\MT@curr@file{\jobname.tex}

```

That was that.

```

3952 </package>

```

```

3953 </package|letterspace>

```

## 15 Configuration Files

Let's now write the font configuration files.

```

3954 <*config>
3955

```

### 15.1 Font Sets

We first declare some sets in the main configuration file.

```

3956 <*m-t>
3957 %%% -----
3958 %%% FONT SETS
3959
3960 \DeclareMicrotypeSet{all}
3961 { }
3962

```

```

3963 \DeclareMicrotypeSet{allmath}
3964 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
3965
3966 \DeclareMicrotypeSet{alltext}
3967 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
3968
3969 \DeclareMicrotypeSet{basicmath}
3970 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
3971   family   = {rm*,sf*},
3972   series    = {md*},
3973   size      = {normalsize,footnotesize,small,large}
3974 }
3975
3976 \DeclareMicrotypeSet{basictext}
3977 { encoding = {OT1,T1,LY1,OT4,QX,T5},
3978   family   = {rm*,sf*},
3979   series    = {md*},
3980   size      = {normalsize,footnotesize,small,large}
3981 }
3982
3983 \DeclareMicrotypeSet{smallcaps}
3984 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3985   shape     = {sc}
3986 }
3987
3988 \DeclareMicrotypeSet{footnotesize}
3989 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3990   size      = {-small}
3991 }
3992
3993 \DeclareMicrotypeSet{scriptsize}
3994 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3995   size      = {-footnotesize}
3996 }
3997
3998 \DeclareMicrotypeSet{normal font}
3999 { font = */*/*/*/* }
4000

```

The default sets.

```

4001 %%% -----
4002 %%% DEFAULT SETS
4003
4004 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4005 \DeclareMicrotypeSetDefault[expansion]{basictext}
4006 \DeclareMicrotypeSetDefault[spacing]{basictext}
4007 \DeclareMicrotypeSetDefault[kerning]{alltext}
4008 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
4009

```

## 15.2 Font Variants and Aliases

```

4010 %%% -----
4011 %%% FONT VARIANTS AND ALIASES
4012

```

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

```

4013 \DeclareMicrotypeVariants{x,j,w,a,d,0,1}
4014

```

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.

```
4015 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
4016 \DeclareMicrotypeAlias{aer}{cmr} % ae
4017 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
4018 \DeclareMicrotypeAlias{cmor}{cmr} % eco
4019 \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).

```
4020 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
4021 \DeclareMicrotypeAlias{qpl}{ppl} % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)
```

The 'FPL Neu' fonts, a 're-implementation' of Palatino.

```
4022 \DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
4023 \DeclareMicrotypeAlias{fp9j}{pplj} % "
4024 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
4025 \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.

```
4026 \DeclareMicrotypeAlias{eur}{eur} % Euler VM
4027 \DeclareMicrotypeAlias{zeus}{eus} % "
```

MicroPress's Charter version (chmath).

```
4028 \DeclareMicrotypeAlias{chr}{bch} % CH Math
```

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```
4029 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
4030 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
4031 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

Euro symbol fonts, to save some files.

```
4032 \DeclareMicrotypeAlias{zpeus}{zpeu} % Adobe Euro sans -> serif
4033 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
4034 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
4035
```

## 15.3 Interaction with babel

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

```
4036 %%% -----
4037 %%% INTERACTION WITH THE `babel' PACKAGE
4038
4039 \DeclareMicrotypeBabelHook
4040 {french,français,acadian,canadien}
4041 {kerning=french, spacing=}
```

```

4042
4043 \DeclareMicrotypeBabelHook
4044   {english,UKenglish,british,USenglish,american}
4045   {kerning=, spacing=nonfrench}
4046
4047 \DeclareMicrotypeBabelHook
4048   {turkish}
4049   {kerning=turkish, spacing=}
4050

```

## 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 L<sup>A</sup>T<sub>E</sub>X 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 Œ for O.

```

4051 </m-t>
4052 <*m-t|zpeu|mys>
4053 %%% -----
4054 %%% CHARACTER INHERITANCE
4055
4056 </m-t|zpeu|mys>
4057 <*m-t>

```

### 15.5.1 OT1

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

```

4058 \DeclareCharacterInheritance
4059 { encoding = OT1 }
4060 { f = {011}, % ff
4061   i = {\i},
4062   j = {\j},
4063   o = {\o},
4064   o = {\o}
4065 }
4066

```

### 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.

```

4067 \DeclareCharacterInheritance
4068 { encoding = T1 }
4069 { A = {\`A,\`A,\^A,\~A,\`A,\r A,\k A,\u A},
4070   a = {\`a,\`a,\^a,\~a,\`a,\r a,\k a,\u a},
4071   C = {\`C,\`C,\v C},
4072   c = {\`c,\`c,\v c},
4073   D = {\v D,\DH},
4074   d = {\v d,\dj},
4075   E = {\`E,\`E,\^E,\`E,\k E,\v E},
4076   e = {\`e,\`e,\^e,\`e,\k e,\v e},
4077   f = {027}, % ff
4078   G = {\u G},
4079   g = {\u g},
4080   I = {\`I,\`I,\^I,\`I,\`I},
4081   i = {\`i,\`i,\^i,\`i,\`i},
4082   j = {\j},
4083   L = {\L,\`L,\v L},
4084   l = {\l,\`l,\v l},
4085   N = {\`N,\~N,\v N},
4086   n = {\`n,\~n,\v n},
4087   O = {\O,\`O,\`O,\^O,\~O,\`O,\H O},
4088   o = {\o,\`o,\`o,\^o,\~o,\`o,\H o},
4089   R = {\`R,\v R},
4090   r = {\`r,\v r},
4091   S = {\`S,\`S,\v S,\SS},
4092   s = {\`s,\`s,\v s},
4093   T = {\c T,\v T},
4094   t = {\c t,\v t},
4095   U = {\`U,\`U,\^U,\`U,\H U,\r U},
4096   u = {\`u,\`u,\^u,\`u,\H u,\r u},
4097   Y = {\`Y,\`Y},
4098   y = {\`y,\`y},
4099   Z = {\`Z,\`Z,\v Z},
4100   z = {\`z,\`z,\v z}

```

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

```

4101 % - = {127},
4102 }
4103

```

### 15.5.3 LY1

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

```

4104 \DeclareCharacterInheritance

```



```

4105 { encoding = LY1 }
4106 { A = {\`A,\`A,\^A,\~A,\"A,\r A},
4107   a = {\`a,\`a,\^a,\~a,\"a,\r a},
4108   C = {\c C},
4109   c = {\c c},
4110   D = {\DH},
4111   E = {\`E,\`E,\^E,\"E},
4112   e = {\`e,\`e,\^e,\"e},
4113   f = {011}, % ff
4114   I = {\`I,\`I,\^I,\"I},
4115   i = {\`i,\`i,\^i,\"i,\i},
4116   L = {\L},
4117   l = {\l},
4118   N = {\~N},
4119   n = {\~n},
4120   O = {\`O,\`O,\^O,\~O,\"O,\O},
4121   o = {\`o,\`o,\^o,\~o,\"o,\o},
4122   S = {\v S},
4123   s = {\v s},
4124   U = {\`U,\`U,\^U,\"U},
4125   u = {\`u,\`u,\^u,\"u},
4126   Y = {\`Y,\"Y},
4127   y = {\`y,\"y},
4128   Z = {\v Z},
4129   z = {\v z}
4130 }
4131

```

#### 15.5.4 OT4

The Polish OT1 extension. More interesting characters here: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

4132 \DeclareCharacterInheritance
4133 { encoding = OT4 }
4134 { A = {\k A},
4135   a = {\k a},
4136   C = {\`C},
4137   c = {\`c},
4138   E = {\k E},
4139   e = {\k e},
4140   f = {011}, % ff
4141   i = {\i},
4142   j = {\j},
4143   L = {\L},
4144   l = {\l},
4145   N = {\`N},
4146   n = {\`n},
4147   O = {\O,\`O},
4148   o = {\o,\`o},
4149   S = {\`S},
4150   s = {\`s},
4151   Z = {\`Z,\.Z},
4152   z = {\`z,\.z}
4153 }
4154

```

### 15.5.5 QX

The Central European QX encoding.<sup>13</sup> Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

4155 \DeclareCharacterInheritance
4156 { encoding = QX }
4157 { A = {\`A,\^A,\~A,\k A,\AA},
4158   a = {\`a,\^a,\~a,\k a,\aa},
4159   C = {\`C,\c C},
4160   c = {\`c,\c c},
4161   D = {\DH},
4162   E = {\`E,\^E,\~E,\k E},
4163   e = {\`e,\^e,\~e,\k e},
4164   f = {011}, % ff
4165   I = {\`I,\^I,\~I,\k I},
4166   i = {\`i,\^i,\~i,\k i,\i},
4167   j = {\j},
4168   L = {\L},
4169   l = {\l},
4170   N = {\`N,\~N},
4171   n = {\`n,\~n},
4172   O = {\`O,\^O,\~O,\k O},
4173   o = {\`o,\^o,\~o,\k o},
4174   S = {\`S,\c S,\v S},
4175   s = {\`s,\c s,\v s},
4176   T = {\c T},
4177   t = {\c t},
4178   U = {\`U,\^U,\~U,\k U},
4179   u = {\`u,\^u,\~u,\k u},
4180   Y = {\`Y,\^Y},
4181   y = {\`y,\^y},
4182   Z = {\`Z,\~Z,\v Z},
4183   z = {\`z,\~z,\v z},
4184   . = \textellipsis
4185 }
4186

```

### 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.

```

4187 \DeclareCharacterInheritance
4188 { encoding = T5 }
4189 { A = {\`A,\^A,\~A,\h A,\d A,\^A,\u A,
4190       \`Acircumflex,\^Acircumflex,\~Acircumflex,\hAcircumflex,\dAcircumflex,
4191       \`Abreve,\^Abreve,\~Abreve,\hAbreve,\dAbreve},
4192   a = {\`a,\^a,\~a,\h a,\d a,\^a,\u a,
4193       \`acircumflex,\^acircumflex,\~acircumflex,\hacircumflex,\dacircumflex,
4194       \`abreve,\^abreve,\~abreve,\h abreve,\d abreve},
4195   D = {\DJ},
4196   d = {\dj},
4197   E = {\`E,\^E,\~E,\h E,\d E,\^E,
4198       \`Ecircumflex,\^Ecircumflex,\~Ecircumflex,\hEcircumflex,\dEcircumflex},
4199   e = {\`e,\^e,\~e,\h e,\d e,\^e,
4200       \`ecircumflex,\^ecircumflex,\~ecircumflex,\hecircumflex,\decircumflex},
4201   I = {\`I,\^I,\~I,\h I,\d I},
4202   i = {\`i,\^i,\~i,\h i,\d i,\i},
4203   O = {\`O,\^O,\~O,\h O,\d O,\^O,\horn O},

```

<sup>13</sup> Thanks to *Maciej Eder*.

```

4204      \0circumflex,\0circumflex,\0circumflex,\h0circumflex,\d0circumflex,
4205      \0horn,\0horn,\0horn,\h0horn,\d0horn},
4206      o = {\0,\o,\-o,\h o,\d o,\^o,\horn o,
4207      \ocircumflex,\ocircumflex,\ocircumflex,\h\ocircumflex,\d\ocircumflex,
4208      \ohorn,\ohorn,\ohorn,\h\ohorn,\d\ohorn},
4209      U = {\U,\'U,\-U,\h U,\d U,\horn U,
4210      \Uhorn,\Uhorn,\Uhorn,\h\Uhorn,\d\Uhorn},
4211      u = {\u,\'u,\-u,\h u,\d u,\horn u,
4212      \uhorn,\uhorn,\uhorn,\h\uhorn,\d\uhorn},
4213      Y = {\Y,\'Y,\-Y,\h Y,\d Y},
4214      y = {\y,\'y,\-y,\h y,\d y}
4215  }
4216
4217  </m-t>

```

### 15.5.7 Euro symbols

Make Euro symbols settings simpler.

```

4218  < *zpeu >
4219  \DeclareCharacterInheritance
4220    { encoding = U,
4221      family   = {zpeu,zpeus,eurosans} }
4222    { E = 128 }
4223
4224  </zpeu >
4225  < *mvs >
4226  \DeclareCharacterInheritance
4227    { encoding = OT1,
4228      family   = mvs }
4229    { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
4230

```

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.

```

4231  \DeclareCharacterInheritance
4232    { encoding = U,
4233      family   = mvs }
4234    { 164 = {099,100,101} }
4235
4236  </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.

```

4237  < *m-t >
4238  %%% -----
4239  %%% TRACKING/LETTERSPACING
4240
4241  \SetTracking
4242    [ name          = default,
4243      no ligatures = {f} ]
4244    { encoding      = {OT1,T1,LY1,OT4,QX} }
4245    { }
4246

```

## 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).

```

4247 %%% -----
4248 %%% EXPANSION
4249
4250 \SetExpansion
4251 [ name = default ]
4252 { encoding = {OT1,OT4,QX,T1,LY1} }
4253 {
4254     A = 500,      a = 700,
4255     \AE = 500,    \ae = 700,
4256     B = 700,      b = 700,
4257     C = 700,      c = 700,
4258     D = 500,      d = 700,
4259     E = 700,      e = 700,
4260     F = 700,
4261     G = 500,      g = 700,
4262     H = 700,      h = 700,
4263     K = 700,      k = 700,
4264     M = 700,      m = 700,
4265     N = 700,      n = 700,
4266     O = 500,      o = 700,
4267     \OE = 500,    \oe = 700,
4268     P = 700,      p = 700,
4269     Q = 500,      q = 700,
4270     R = 700,
4271     S = 700,      s = 700,
4272     U = 700,      u = 700,
4273     W = 700,      w = 700,
4274     Z = 700,      z = 700,
4275     2 = 700,
4276     3 = 700,
4277     6 = 700,
4278     8 = 700,
4279     9 = 700
4280 }
4281

```

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

```

4282 \SetExpansion
4283 [ name = T5 ]
4284 { encoding = T5 }
4285 {
4286     A = 500,      a = 700,
4287     B = 700,      b = 700,
4288     C = 700,      c = 700,
4289     D = 500,      d = 700,
4290     E = 700,      e = 700,
4291     F = 700,
4292     G = 500,      g = 700,
4293     H = 700,      h = 700,
4294     K = 700,      k = 700,
4295     M = 700,      m = 700,
4296     N = 700,      n = 700,
4297     O = 500,      o = 700,
4298     P = 700,      p = 700,
4299     Q = 500,      q = 700,
4300     R = 700,
4301     S = 700,      s = 700,
4302     U = 700,      u = 700,

```

```

4303     W = 700,      w = 700,
4304     Z = 700,      z = 700,
4305     2 = 700,
4306     3 = 700,
4307     6 = 700,
4308     8 = 700,
4309     9 = 700
4310   }
4311
4312 </m-t>

```

## 15.8 Character Protrusion

```

4313 %%% -----
4314 %%% PROTRUSION
4315

```

For future historians, Hàn Thê 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.

```

4316 <*cfg-t>
4317 \SetProtrusion
4318 <m-t> [ name      = default ]

```

We also create configuration files for the fonts Bitstream Charter (NFSS code `bch`),

```

4319 <bch> [ name      = bch-default ]

```

Bitstream Letter Gothic (`blg`),

```

4320 <blg> [ name      = blg-default ]
      Computer Modern Roman (cmr),
4321 <cmr> [ name      = cmr-default ]
      Adobe Garamond (pad, padx, padj),
4322 <pad> [ name      = pad-default ]
      Minion14 (pmnx, pmnj),
4323 <pmn> [ name      = pmnj-default ]
      Palatino (ppl, pplx, pplj),
4324 <ppl> [ name      = ppl-default ]
      Times (ptm, ptmx, ptmj)
4325 <ptm> [ name      = ptm-default ]
      and URW Garamond (ugm).
4326 <ugm> [ name      = ugm-default ]
4327 <m-t> { encoding = OT1      }
4328 <cmr> { }
4329 <bch|blg|pad|pmn|ugm> { encoding = OT1,
4330 <ppl|ptm> { encoding = {OT1,OT4},
4331 <bch>      family   = bch }
4332 <blg>      family   = blg }
4333 <pad>      family   = {pad,padx,padj} }
4334 <pmn>      family   = pmnj }
4335 <ppl>      family   = {ppl,pplx,pplj} }
4336 <ptm>      family   = {ptm,ptmx,ptmj} }
4337 <ugm>      family   = ugm }
4338 {
4339 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm>      A = {50,50},
4340 <ugm>      A = {50,100},
4341 <m-t|pad|ptm>      \AE = {50, },
4342 <ugm>      \AE = {150,50},
4343 <ugm>      B = { ,50},
4344 <bch|pad|pmn|ugm>      C = {50, },
4345 <bch|pad|pmn>      D = { ,50},
4346 <ugm>      D = { ,70},
4347 <ugm>      E = { ,50},
4348 <m-t|bch|cmr|pad|pmn|ptm>      F = { ,50},
4349 <ugm>      F = { ,70},
4350 <bch|pad|pmn>      G = {50, },
4351 <ugm>      G = {50,50},
4352 <blg>      I = {150,150},
4353 <m-t|cmr|pad|pmn|ppl|ptm|ugm>      J = {50, },
4354 <bch|blg>      J = {100, },
4355 <!blg>      K = { ,50},
4356 <blg>      K = {50, },
4357 <m-t|bch|cmr|pad|pmn|ppl>      L = { ,50},
4358 <blg>      L = { ,150},
4359 <ptm>      L = { ,80},
4360 <ugm>      L = { ,120},
4361 <bch|pad|pmn|ugm>      O = {50,50},
4362 <pad|pmn>      \OE = {50, },
4363 <ugm>      \OE = {50,50},
4364 <blg>      P = { ,100},
4365 <ugm>      P = { ,50},
4366 <bch|pad|pmn>      Q = {50,70},
4367 <ugm>      Q = {50,50},
4368 <bch>      R = { ,50},

```

---

14 Contributed by Harald Harders ([h.harders@tu-bs.de](mailto:h.harders@tu-bs.de)).

```

4369 <ugm>      R = { ,70},
4370 <m-t|bch|cmr|pad|pmn|ppl|ptm>    T = {50,50},
4371 <blg>      T = {100,100},
4372 <ugm>      T = {70,70},
4373 <m-t|bch|cmr|pad|pmn|ppl|ptm>    V = {50,50},
4374 <blg|ugm>    V = {70,70},
4375 <m-t|bch|cmr|pad|pmn|ppl|ptm>    W = {50,50},
4376 <ugm>      W = {70,70},
4377 <m-t|bch|cmr|pad|pmn|ppl|ptm>    X = {50,50},
4378 <ugm>      X = {50,70},
4379 <m-t|bch|cmr|pad|pmn|ppl>      Y = {50,50},
4380 <blg|ptm|ugm>    Y = {80,80},
4381 <ugm>      Z = {50,50},
4382 <blg>      f = {150,100},
4383 <blg>      i = {150,150},
4384 <blg>      j = {100,100},
4385 <m-t|bch|cmr|pad|pmn|ppl|ptm>    k = { ,50},
4386 <ugm>      k = { ,70},
4387 <blg>      l = {150,150},
4388 <pmn>      l = { ,50},
4389 <pad|ppl>    p = {50,50},
4390 <ugm>      p = { ,50},
4391 <pad|ppl>    q = {50, },
4392 <!blg>      r = { ,50},
4393 <blg>      r = {100, 80},
4394 <cmr|pad|pmn>    t = { ,70},
4395 <bch>      t = { ,50},
4396 <blg>      t = {150, 80},
4397 <ugm>      t = { ,100},
4398 <m-t|bch|cmr|pad|pmn|ppl|ptm>    v = {50,50},
4399 <blg>      v = {100,100},
4400 <ugm>      v = {50,70},
4401 <m-t|bch|cmr|pad|pmn|ppl|ptm>    w = {50,50},
4402 <ugm>      w = {50,70},
4403 <!blg>      x = {50,50},
4404 <blg>      x = {100,100},
4405 <m-t|bch|pad|pmn>    y = { ,50},
4406 <blg>      y = { 50,100},
4407 <cmr|ppl|ptm>    y = {50,70},
4408 <ugm>      y = { ,70},

4409 <cmr>      0 = { ,50},
4410 <m-t>      1 = {50,50},
4411 <bch|blg|pad|ptm|ugm>    1 = {150,150},
4412 <cmr>      1 = {100,200},
4413 <pmn>      1 = { ,50},
4414 <ppl>      1 = {100,100},
4415 <bch|cmr|pad|ugm>    2 = {50,50},
4416 <blg>      2 = { ,100},
4417 <bch|pmn>    3 = {50, },
4418 <cmr|pad|ugm>    3 = {50,50},
4419 <blg>      3 = {100, },
4420 <m-t|pad>    4 = {50,50},
4421 <bch>      4 = {100,50},
4422 <blg>      4 = {100, },
4423 <cmr|ugm>    4 = {70,70},
4424 <pmn>      4 = {50, },
4425 <ptm>      4 = {70, },
4426 <cmr>      5 = { ,50},
4427 <pad>      5 = {50,50},
4428 <bch>      6 = {50, },
4429 <cmr>      6 = { ,50},
4430 <pad>      6 = {50,50},

```

```

4431 <m-t>      7 = {50,50},
4432 <bch|pad|pmn|ugm>      7 = {50,80},
4433 <blg>      7 = {100,100},
4434 <cmr|ptm>      7 = {50,100},
4435 <ppl>      7 = { ,50},
4436 <cmr>      8 = { ,50},
4437 <bch|pad>      9 = {50,50},
4438 <cmr>      9 = { ,50},
4439 <m-t|cmr|pad|pmn|ppl|ptm|ugm>      . = { ,700},
4440 <bch>      . = { ,600},
4441 <blg>      . = {400,500},
4442 <!blg>      {,}= { ,500},
4443 <blg>      {,}= {300,400},
4444 <m-t|cmr|pad|pmn|ppl|ptm|ugm>      : = { ,500},
4445 <bch>      : = { ,400},
4446 <blg>      : = {300,400},
4447 <m-t|bch|pad|pmn|ptm>      ; = { ,300},
4448 <blg>      ; = {200,300},
4449 <cmr|ppl>      ; = { ,500},
4450 <ugm>      ; = { ,400},
4451 <!blg>      ! = { ,100},
4452 <blg>      ! = {200,200},
4453 <m-t|pad|pmn|ptm>      ? = { ,100},
4454 <bch|cmr|ppl|ugm>      ? = { ,200},
4455 <blg>      ? = {150,150},
4456 <pmn>      " = {300,300},
4457 <m-t|bch|cmr|pad|pmn|ppl>      @ = {50,50},
4458 <ptm>      @ = {100,100},
4459 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm>      ~ = {200,250},
4460 <ugm>      ~ = {300,350},
4461 <pad|ppl|ptm>      & = {50,100},
4462 <ugm>      & = { ,100},
4463 <m-t|cmr|pad|pmn>      \% = {50,50},
4464 <bch>      \% = { ,50},
4465 <ppl|ptm>      \% = {100,100},
4466 <ugm>      \% = {50,100},
4467 <blg>      \# = {100,100},
4468 <m-t|ppl|ptm|ugm>      * = {200,200},
4469 <bch|pmn>      * = {200,300},
4470 <blg>      * = {150,200},
4471 <cmr|pad>      * = {300,300},
4472 <m-t|cmr|ppl|ptm>      + = {250,250},
4473 <bch>      + = {150,250},
4474 <pad>      + = {300,300},
4475 <blg|pmn>      + = {150,200},
4476 <ugm>      + = {250,300},
4477 <blg|ugm>      {=}= {200,200},
4478 <m-t|pad|pmn|ptm>      ( = {100, }, ) = { ,200},
4479 <bch|ugm>      ( = {200, }, ) = { ,200},
4480 <blg>      ( = {300, }, ) = { ,300},
4481 <cmr|ppl>      ( = {100, }, ) = { ,300},
4482 <bch|pmn>      [ = {100, }, ] = { ,100},
4483 <blg>      [ = {300,100}, ] = { ,300},

4484 <m-t|pad|pmn|ptm>      / = {100,200},
4485 <bch>      / = { ,200},
4486 <blg>      / = {300,300},
4487 <cmr|ppl>      / = {200,300},
4488 <ugm>      / = {100,300},
4489 <m-t|ptm>      - = {500,500},
4490 <bch|cmr|ppl>      - = {400,500},
4491 <blg>      - = {300,400},
4492 <pad>      - = {300,500},

```



```

4493 <pmn>      - = {200,400},
4494 <ugm>      - = {500,600},
4495 <blg>      < = {200,100},    > = {100,200},
4496 <blg>      - = {150,250},
4497 <blg>      | = {250,250},
4498 <m-t|pmn>   \textendash      = {200,200},    \textendash      = {150,150},
4499 <bch>        \textendash      = {200,300},    \textendash      = {150,250},
4500 <cmr>        \textendash      = {400,300},    \textendash      = {300,200},
4501 <pad|ppl|ptm> \textendash      = {300,300},    \textendash      = {200,200},
4502 <ugm>        \textendash      = {250,300},    \textendash      = {250,250},

```

Why settings for left *and* right quotes? Because in some languages they might be used like that (see the csquotes package for examples).

```

4503 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
4504 <blg>         \textquoteleft = {400,600}, \textquoteright = {400,600},
4505 <cmr>         \textquoteleft = {500,700}, \textquoteright = {500,600},
4506 <pad|ppl>     \textquoteleft = {500,700}, \textquoteright = {500,700},
4507 <ptm>         \textquoteleft = {500,500}, \textquoteright = {300,500},
4508 <ugm>         \textquoteleft = {300,600}, \textquoteright = {300,600},
4509 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300}
4510 <blg>         \textquotedblright = {300,400}
4511 <cmr>         \textquotedblleft = {500,300}, \textquotedblright = {200,600}
4512 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4513 <ugm>         \textquotedblleft = {400,400}, \textquotedblright = {400,400}
4514 }
4515

```

Greek uppercase letters are in OT1 encoding only.

```

4516 <*cmr>
4517 \SetProtrusion
4518 [ name      = cmr-OT1,
4519   load      = cmr-default ]
4520 { encoding = {OT1,OT4},
4521   family   = cmr }
4522 {
4523   \AE = { 50,   },
4524   "00 = {   ,150}, % \Gamma
4525   "01 = {100,100}, % \Delta
4526   "02 = { 50, 50}, % \Theta
4527   "03 = {100,100}, % \Lambda
4528   "06 = { 50, 50}, % \Sigma
4529   "07 = {100,100}, % \Upsilon
4530   "08 = { 50, 50}, % \Phi
4531   "09 = { 50, 50} % \Psi

```

Remaining slots can be found in the source file.

```

4532 }
4533
4534 </cmr>

```

T1 and LY1 encodings contain some more characters. The default list will be loaded first.

```

4535 \SetProtrusion
4536 <m-t> [ name      = T1-default,
4537 <bch> [ name      = bch-T1,
4538 <blg> [ name      = blg-T1,
4539 <cmr> [ name      = cmr-T1,
4540 <pad> [ name      = pad-T1,
4541 <pmn> [ name      = pmnj-T1,
4542 <ppl> [ name      = ppl-T1,
4543 <ptm> [ name      = ptm-T1,
4544 <ugm> [ name      = ugm-T1,

```

```

4545 <m-t>      load      = default      ]
4546 <bch>      load      = bch-default ]
4547 <blg>      load      = blg-default  ]
4548 <cmr>      load      = cmr-default  ]
4549 <pad>      load      = pad-default  ]
4550 <pmn>      load      = pmnj-default ]
4551 <ppl>      load      = ppl-default  ]
4552 <ptm>      load      = ptm-default  ]
4553 <ugm>      load      = ugm-default  ]
4554 <m-t>      { encoding = {T1,LY1}    }
4555 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4556 <blg|ptm|ugm> { encoding = {T1},
4557 <bch>      family    = bch }
4558 <blg>      family    = blg }
4559 <cmr>      family    = cmr }
4560 <pad>      family    = {pad,padx,padj} }
4561 <pmn>      family    = pmnj }
4562 <ppl>      family    = {ppl,pplx,pplj} }
4563 <ptm>      family    = {ptm,ptmx,ptmj} }
4564 <ugm>      family    = ugm }
4565 {
4566 <cmr>      \AE = {50, },
4567 <bch>      \OE = {50, },
4568 <pmn>      \TH = { ,50},
4569 <blg>      \v L = { ,250},
4570 <blg>      \v d = { ,250},
4571 <blg>      \v l = { ,250},
4572 <blg>      \v t = { ,250},
4573 <blg>      127 = {300,400},
4574 <blg>      156 = {100, }, % IJ
4575 <blg>      188 = { 80, 80}, % ij
4576 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
4577 <cmr>      _ = {200,200},
4578 <ugm>      _ = {100,200},
4579 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
4580 <bch>      \textbackslash = {150,200},
4581 <blg>      \textbackslash = {250,300},
4582 <cmr|ppl> \textbackslash = {200,300},
4583 <ugm>      \textbackslash = {100,300},
4584 <ugm>      \textbar      = {200,200},
4585 <blg>      \textendash    = {300,300}, \textemdash      = {150,150},
4586 <cmr>      \textquotedbl = {300,400}, \textquotedblleft = {300,400},
4587 <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.

```

4588 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4589 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
4590 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4591 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsingright = {300,400},
4592 <blg> \guilsinglleft = {300,500}, \guilsingright = {300,500},
4593 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsingright = {300,500},
4594 <ugm> \guilsinglleft = {400,400}, \guilsingright = {300,600},
4595 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
4596 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
4597 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
4598 <blg|pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
4599 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4600 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
4601 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
4602 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4603 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},

```

```

4604 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
4605 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
4606 <pmn> \textless = {100, }, \textgreater = { ,100},
4607 <pmn> \textvisiblespace = {100,100} % not in LY1
4608 }
4609

```

The lmodern fonts, on the other hand, restore the original kerning from the OT1 fonts, and so do we. Silly, isn't it?

```

4610 <*cmr>
4611 \SetProtrusion
4612 [ name = lmr-T1,
4613   load = cmr-T1 ]
4614 { encoding = {T1,LY1},
4615   family = lmr }
4616 {
4617   \textquotedblleft = {500,300},
4618   \quotedblbase = {500,300}
4619 }
4620
4621 </cmr>

```

Settings for the QX encoding (generic and Times). It also includes some glyphs otherwise in TS1.

```

4622 <*m-t|ptm>
4623 \SetProtrusion
4624 <m-t> [ name = QX-default,
4625 <ptm> [ name = ptm-QX,
4626 <m-t>   load = default ]
4627 <ptm>   load = ptm-default ]
4628 <m-t> { encoding = QX }
4629 <ptm> { encoding = QX,
4630 <ptm>   family = {ptm,ptmx,ptmj} }
4631 {
4632 <ptm>   * = {200,200},
4633   {=} = {100,100},
4634   \textunderscore = {100,100},
4635   \textbackslash = {100,200},
4636   \quotedblbase = {400,400},
4637 <m-t>   \guillemotleft = {200,200}, \guillemotright = {200,200},
4638 <ptm>   \guillemotleft = {300,300}, \guillemotright = {200,400},
4639   \textexclamdown = {100, }, \textquestiondown = {100, },
4640 <m-t>   \textbraceleft = {400,200}, \textbraceright = {200,400},
4641 <ptm>   \textbraceleft = {200,200}, \textbraceright = {200,300},
4642   \textless = {200,100}, \textgreater = {100,200},
4643   \textminus = {200,200}, \textdegree = {300,300},
4644 <m-t>   \copyright = {100,100}, \textregistered = {100,100}
4645 <ptm>   \copyright = {100,150}, \textregistered = {100,150},
4646 <ptm>   \textxgeq = { ,100}, \textxleq = {100, },
4647 <ptm>   \textalpha = { , 50}, \textDelta = { 70, 70},
4648 <ptm>   \textpi = { 50, 80}, \textSigma = { , 70},
4649 <ptm>   \textmu = { , 80}, \texteuro = { 50, 50},
4650 <ptm>   \textellipsis = {150,200}, \textasciitilde = { 80, 80},
4651 <ptm>   \textapprox = { 50, 50}, \textinfty = {100,100},
4652 <ptm>   \textdagger = {150,150}, \textdaggerdbl = {100,100},
4653 <ptm>   \textdiv = { 50,150}, \textsection = { 80, 80},
4654 <ptm>   \texttimes = {100,150}, \textpm = { 50, 80},
4655 <ptm>   \textbullet = {150,150}, \textperiodcentered = {300,300},
4656 <ptm>   \textquotesingle = {500,500}, \textquotedbl = {300,300},
4657 <ptm>   \textperthousand = { ,50}
4658 }

```

4659  
4660 *</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.

```

4661 <*cmr|bch>
4662 \SetProtrusion
4663 <cmr> [ name = cmr-T5,
4664 <cmr> load = cmr-default ]
4665 <bch> [ name = bch-T5,
4666 <bch> load = bch-default ]
4667 { encoding = T5,
4668 <cmr> family = cmr }
4669 <bch> family = bch }
4670 {
4671 <bch> _ = {100,100},
4672 <bch> \textbackslash = {150,200},
4673 <cmr> \textbackslash = {200,300},
4674 <cmr> \textquotedblleft = {200,600},
4675 <cmr> \textquotedbl = {300,300},
4676 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4677 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4678 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4679 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4680 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},
4681 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
4682 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
4683 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
4684 \textless = {200,100}, \textgreater = {100,200}
4685 }
4686
4687 </cmr|bch>
4688 <*pmn>
4689 \SetProtrusion
4690 [ name = pmnx-OT1,
4691 load = pmnj-default ]
4692 { encoding = OT1,
4693 family = pmnx }
4694 {
4695 1 = {230,180}
4696 }
4697
4698 \SetProtrusion
4699 [ name = pmnx-T1,
4700 load = pmnj-T1 ]
4701 { encoding = {T1,LY1},
4702 family = pmnx }
4703 {
4704 1 = {230,180}
4705 }
4706
4707 </pmn>

```

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

```

4708 <*ptm>
4709 \SetProtrusion
4710 [ name = ptm-LY1,
4711 load = ptm-T1 ]
4712 { encoding = LY1,
4713 family = {ptm,ptmx,ptmj} }
4714 {

```

```

4715      _ = {100,100},
4716      \texttrademark = {100,100},
4717      \textregistered = {100,100},
4718      \textcopyright = {100,100},
4719      \textdegree = {300,300},
4720      \textminus = {200,200},
4721      \textellipsis = {150,200},
4722      \texteuro = { , }, % ?
4723      \textcent = {100,100},
4724      \textquotesingle = {500,500},
4725      \textflorin = { 50, 70},
4726      \textdagger = {150,150},
4727      \textdaggerdbl = {100,100},
4728      \textperthousand = { , 50},
4729      \textbullet = {150,150},
4730      \textonesuperior = {100,100},
4731      \texttwosuperior = { 50, 50},
4732      \textthreesuperior = { 50, 50},
4733      \textperiodcentered = {300,300},
4734      \textplusminus = { 50, 80},
4735      \textmultiply = {100,100},
4736      \textdivide = { 50,150}

```

Remaining slots in the source file.

```

4737    }
4738
4739  </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.

```

4740 \SetProtrusion
4741 <m-t> [ name = OT1-it ]
4742 <bch> [ name = bch-it ]
4743 <blg> [ name = blg-it,
4744 <blg> load = blg-default ]
4745 <cmr> [ name = cmr-it ]
4746 <pad> [ name = pad-it ]
4747 <pmn> [ name = pmn-it ]
4748 <ppl> [ name = ppl-it ]
4749 <ptm> [ name = ptm-it ]
4750 <ugm> [ name = ugm-it ]
4751 <m-t|bch|blg|pad|pmn|ugm> { encoding = OT1,
4752 <ppl|ptm> { encoding = {OT1,OT4},
4753 <bch> family = bch,
4754 <blg> family = blg,
4755 <pad> family = {pad,padx,padj},
4756 <pmn> family = pmnj,
4757 <ppl> family = {ppl,pplx,pplj},
4758 <ptm> family = {ptm,ptmx,ptmj},
4759 <ugm> family = ugm,
4760 <m-t|bch|pad|pmn|ppl|ptm> shape = {it,sl} }
4761 <blg|ugm> shape = it }
4762 <cmr> { }
4763 {
4764 <cmr|ptm> A = {100,50},
4765 <pad|pmn> A = {50, },
4766 <ugm> A = { ,150},
4767 <ppl> A = {50,50},

```

```

4768 <ptm> \AE = {100, },
4769 <pad|ppl> \AE = {50, },
4770 <pmn> \AE = { , -50},
4771 <cmr|pad|ppl|ptm> B = {50, },
4772 <pmn> B = {20, -50},
4773 <bch|ppl|ptm|ugm> C = {50, },
4774 <cmr|pad> C = {100, },
4775 <pmn> C = {50, -50},
4776 <cmr|pad|ppl|ptm> D = {50, 50},
4777 <pmn> D = {20, },
4778 <cmr|pad|ppl|ptm> E = {50, },
4779 <pmn> E = {20, -50},
4780 <cmr|pad|ptm> F = {100, },
4781 <pmn> F = {10, },
4782 <ppl> F = {50, },
4783 <bch|ppl|ptm|ugm> G = {50, },
4784 <cmr|pad> G = {100, },
4785 <pmn> G = {50, -50},
4786 <cmr|pad|ppl|ptm> H = {50, },
4787 <cmr|pad|ptm> I = {50, },
4788 <pmn> I = {20, -50},
4789 <cmr|ptm> J = {100, },
4790 <pad> J = {50, },
4791 <pmn> J = {20, },
4792 <cmr|pad|ppl|ptm> K = {50, },
4793 <pmn> K = {20, },
4794 <cmr|pad|ppl|ptm> L = {50, },
4795 <pmn> L = {20, 50},
4796 <ugm> L = { , 100},
4797 <cmr|ptm> M = {50, },
4798 <pmn> M = { , -30},
4799 <cmr|ptm> N = {50, },
4800 <pmn> N = { , -30},
4801 <bch|pmn|ppl|ptm> O = {50, },
4802 <cmr|pad> O = {100, },
4803 <ugm> O = {70, 50},
4804 <pmn|ppl|ptm> \OE = {50, },
4805 <pad> \OE = {100, },
4806 <cmr|pad|ppl|ptm> P = {50, },
4807 <pmn> P = {20, -50},
4808 <bch|pmn|ppl|ptm> Q = {50, },
4809 <cmr|pad> Q = {100, },
4810 <ugm> Q = {70, 50},
4811 <cmr|pad|ppl|ptm> R = {50, },
4812 <pmn> R = {20, },
4813 <bch|cmr|pad|ppl|ptm> S = {50, },
4814 <pmn> S = {20, -30},
4815 <bch|cmr|pad|ppl|ptm> $ = {50, },
4816 <pmn> $ = {20, -30},
4817 <bch|pmn|ugm> T = {70, },
4818 <cmr|pad|ppl|ptm> T = {100, },
4819 <cmr|pad|ppl|ptm> U = {50, },
4820 <pmn> U = {50, -50},
4821 <cmr|pad|pmn|ugm> V = {100, },
4822 <ppl|ptm> V = {100, 50},
4823 <cmr|pad|pmn|ugm> W = {100, },
4824 <ppl> W = {50, },
4825 <ptm> W = {100, 50},
4826 <cmr|ppl|ptm> X = {50, },
4827 <cmr|ptm> Y = {100, },
4828 <pmn> Y = {50, },
4829 <ppl> Y = {100, 50},
4830 <pmn> Z = { , -50},

```

```

4831 <pmn>      d = { , -50},
4832 <pad|pmn>   f = { , -100},
4833 <pmn>      i = { , -30},
4834 <pmn>      j = { , -30},
4835 <pmn>      l = { , -100},
4836 <bch>      o = {50,50},
4837 <bch>      p = { , 50},
4838 <pmn>      p = {-50, },
4839 <bch>      q = {50, },
4840 <pmn>      r = { , 50},
4841 <bch>      t = { , 50},
4842 <pmn|ugm>   v = {50, },
4843 <bch>      w = { , 50},
4844 <pmn|ugm>   w = {50, },
4845 <bch>      y = { , 50},
4846 <cmr>      0 = {100, },
4847 <bch|ptm>   1 = {150,100},
4848 <cmr>      1 = {200,50},
4849 <pad>      1 = {150, },
4850 <pmn>      1 = {50, },
4851 <ppl>      1 = {100, },
4852 <ugm>      1 = {150,150},
4853 <cmr>      2 = {100,-100},
4854 <pad|ppl|ptm> 2 = {50, },
4855 <pmn>      2 = {-50, },
4856 <bch>      3 = {50, },
4857 <cmr>      3 = {100,-100},
4858 <pmn>      3 = {-100, },
4859 <ptm>      3 = {100,50},
4860 <bch>      4 = {100, },
4861 <cmr|pad>   4 = {150, },
4862 <ppl|ptm>   4 = {50, },
4863 <cmr>      5 = {100, },
4864 <ptm>      5 = {50, },
4865 <bch>      6 = {50, },
4866 <cmr>      6 = {100, },
4867 <bch|pad|ptm> 7 = {100, },
4868 <cmr>      7 = {200,-150},
4869 <pmn>      7 = {20, },
4870 <ppl>      7 = {50, },
4871 <cmr>      8 = {50,-50},
4872 <cmr>      9 = {100,-100},
4873 <m-t|cmr|pad|pmn|ppl> . = { , 500},
4874 <blg>      . = {400,600},
4875 <bch|ptm|ugm> . = { , 700},
4876 <blg>      {,}= {300,500},
4877 <m-t|cmr|pad|pmn|ppl> {,}= { , 500},
4878 <bch|ugm>   {,}= { , 600},
4879 <ptm>      {,}= { , 700},
4880 <m-t|cmr|pad|ppl> : = { , 300},
4881 <bch|ugm>   : = { , 400},
4882 <pmn>      : = { , 200},
4883 <ptm>      : = { , 500},
4884 <m-t|cmr|pad|ppl> ; = { , 300},
4885 <bch|ugm>   ; = { , 400},
4886 <pmn>      ; = { , 200},
4887 <ptm>      ; = { , 500},
4888 <ptm>      ! = { , 100},
4889 <bch>      ? = { , 200},
4890 <ptm>      ? = { , 100},
4891 <ppl>      ? = { , 300},
4892 <pmn>      " = {400,200},
4893 <m-t|pad|pmn|ppl|ptm> & = {50,50},

```

```

4894 <bch>      & = { ,80},
4895 <cmr>      & = {100,50},
4896 <ugm>      & = {50,100},
4897 <m-t|cmr|pad|pmn> \% = {100, },
4898 <bch>      \% = {50,50},
4899 <ppl|ptm>   \% = {100,100},
4900 <ugm>      \% = {100,50},
4901 <m-t|pmn|ppl> * = {200,200},
4902 <bch>      * = {300,200},
4903 <cmr>      * = {400,100},
4904 <pad>      * = {500,100},
4905 <ptm|ugm>   * = {400,200},
4906 <m-t|cmr|pmn|ppl> + = {150,200},
4907 <bch|ugm>   + = {250,250},
4908 <pad|ptm>   + = {250,200},
4909 <m-t|pad|pmn|ppl> @ = {50,50},
4910 <bch>      @ = {80,50},
4911 <cmr>      @ = {200,50},
4912 <ptm>      @ = {150,150},
4913 <m-t|bch|ugm> ~ = {150,150},
4914 <cmr|pad|pmn|ppl|ptm> ~ = {200,150},
4915 <ugm>      {=} = {200,200},
4916 <!blg>     ( = {200, }, ) = { ,200},
4917 <m-t|cmr|pad|ppl|ptm|ugm> / = {100,200},
4918 <bch>      / = { ,150},
4919 <pmn>      / = {100,150},
4920 <m-t>      - = {300,300},
4921 <bch|pad>   - = {300,400},
4922 <pmn>      - = {200,300},
4923 <cmr>      - = {500,300},
4924 <ppl>      - = {300,500},
4925 <ptm>      - = {500,500},
4926 <ugm>      - = {400,700},
4927 <blg>     - = {0,300},
4928 <m-t|pmn>   \textendash = {200,200}, \textendash = {150,150},
4929 <bch>      \textendash = {200,300}, \textendash = {150,200},
4930 <cmr>      \textendash = {500,300}, \textendash = {400,200},
4931 <pad|ppl|ptm|ugm> \textendash = {300,300}, \textendash = {200,200},
4932 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
4933 <blg>      \textquoteleft = {400,400}, \textquoteright = {400,400},
4934 <cmr|pad>   \textquoteleft = {800,200}, \textquoteright = {800,200},
4935 <ppl>      \textquoteleft = {700,400}, \textquoteright = {700,400},
4936 <ptm>      \textquoteleft = {800,500}, \textquoteright = {800,500},
4937 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
4938 <blg>      \textquotedblright = {300,300}
4939 <cmr>      \textquotedblleft = {700,100}, \textquotedblright = {500,300}
4940 <pad>      \textquotedblleft = {700,200}, \textquotedblright = {700,200}
4941 <ppl>      \textquotedblleft = {500,300}, \textquotedblright = {500,300}
4942 <ptm>      \textquotedblleft = {700,400}, \textquotedblright = {700,400}
4943 <ugm>      \textquotedblleft = {600,200}, \textquotedblright = {600,200}
4944 }
4945
4946 <*cmr>
4947 \SetProtrusion
4948 [ name = cmr-it-OT1,
4949   load = cmr-it ]
4950 { encoding = {OT1,OT4},
4951   family = cmr,
4952   shape = it }
4953 {
4954   \AE = {100, },
4955   \OE = {100, },
4956   "00 = {200,150}, % \Gamma

```



```

4957 "01 = {150,100}, % \Delta
4958 "02 = {150, 50}, % \Theta
4959 "03 = {150, 50}, % \Lambda
4960 "04 = {100,100}, % \Xi
4961 "05 = {100,100}, % \Pi
4962 "06 = {100, 50}, % \Sigma
4963 "07 = {200,150}, % \Upsilon
4964 "08 = {150, 50}, % \Phi
4965 "09 = {150,100}, % \Psi
4966 "0A = { 50, 50} % \Omega
4967 }
4968
4969 </cmr>
4970 \SetProtrusion
4971 <m-t> [ name      = Tl-it-default,
4972 <bch> [ name      = bch-it-Tl,
4973 <blg> [ name      = blg-it-Tl,
4974 <cmr> [ name      = cmr-it-Tl,
4975 <pad> [ name      = pad-it-Tl,
4976 <pmn> [ name      = pmnj-it-Tl,
4977 <ppl> [ name      = ppl-it-Tl,
4978 <ptm> [ name      = ptm-it-Tl,
4979 <ugm> [ name      = ugm-it-Tl,
4980 <m-t> load      = OTl-it ]
4981 <bch> load      = bch-it ]
4982 <blg> load      = blg-Tl ]
4983 <cmr> load      = cmr-it ]
4984 <pmn> load      = pmnj-it ]
4985 <pad> load      = pad-it ]
4986 <ppl> load      = ppl-it ]
4987 <ptm> load      = ptm-it ]
4988 <ugm> load      = ugm-it ]
4989 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {Tl,Ly1},
4990 <blg|ptm|ugm> { encoding = {Tl},
4991 <bch> family    = bch,
4992 <blg> family    = blg,
4993 <cmr> family    = cmr,
4994 <pmn> family    = pmnj,
4995 <pad> family    = {pad,padx,padj},
4996 <ppl> family    = {ppl,pplx,pplj},
4997 <ptm> family    = {ptm,ptmx,ptmj},
4998 <ugm> family    = ugm,
4999 <m-t|bch|pad|pmn|ppl|ptm> shape    = {it,sl} }
5000 <blg|cmr|ugm> shape    = it }
5001 {
5002 <m-t|bch|pmn> _ = { ,100},
5003 <blg> _ = {0,300},
5004 <cmr|ugm> _ = {100,200},
5005 <pad|ppl|ptm> _ = {100,100},
5006 <blg> . = {400,600},
5007 <blg> {,}= {300,500},
5008 <cmr> \AE = {100, },
5009 <bch> \OE = { 50, },
5010 <cmr> \OE = {100, },
5011 <pmn> 031 = { , -100}, % ffl
5012 <cmr|ptm> 156 = {100, }, % IJ
5013 <pad> 156 = {50, }, % IJ
5014 <pmn> 156 = {20, }, % IJ
5015 <pmn> 188 = { , -30}, % ij
5016 <pmn> \v t = { ,100},
5017 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
5018 <cmr|ugm> \textbackslash = {300,300},
5019 <bch> \textbackslash = {150,150},

```

```

5020 <pmn> \textbackslash = {100,150},
5021 <ugm> \textbar = {200,200},
5022 <cmr> \textquotedblleft = {500,300},
5023 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
5024 <blg> \textquotedbl = {300,300}, \textquotedblleft = {300,300},
5025 <blg> \textquotedblright = {300,300}, \quotedblbase = {200,600},
5026 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
5027 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5028 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
5029 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
5030 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
5031 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5032 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5033 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5034 <pad> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
5035 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
5036 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
5037 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
5038 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
5039 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
5040 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
5041 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
5042 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
5043 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
5044 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
5045 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
5046 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
5047 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
5048 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
5049 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
5050 <pmn> \textvisiblespace = {100,100}
5051 }
5052
5053 <*m-t|ptm>
5054 \SetProtrusion
5055 <m-t> [ name = QX-it-default,
5056 <ptm> [ name = ptm-it-QX,
5057 <m-t> load = OT1-it ]
5058 <ptm> load = ptm-it ]
5059 { encoding = {QX},
5060 <ptm> family = {ptm,ptmx,ptmj},
5061 shape = {it,sl} }
5062 {
5063 <ptm> 009 = { , 50}, % fk
5064 {=} = {100,100},
5065 <m-t> \textunderscore = {100,100},
5066 <ptm> \textunderscore = {100,150},
5067 \textbackslash = {100,200},
5068 \quotedblbase = {300,400},
5069 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
5070 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
5071 \textexclamdown = {200, }, \textquestiondown = {200, },
5072 \textbraceleft = {200,100}, \textbraceright = {200,200},
5073 \textless = {100,100}, \textgreater = {100,100},
5074 \textminus = {200,200}, \textdegree = {300,150},
5075 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5076 <ptm> \textregistered = {100,150}, \copyright = {100,150},
5077 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
5078 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
5079 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
5080 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
5081 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
5082 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},

```

```

5083 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5084 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
5085 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
5086 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
5087 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5088 <ptm> \textperthousand = { ,50}
5089 }
5090
5091 </m-t|ptm>
5092 <*cmr|bch>
5093 \SetProtrusion
5094 <cmr> [ name = cmr-it-T5,
5095 <cmr> load = cmr-it ]
5096 <bch> [ name = bch-it-T5,
5097 <bch> load = bch-it ]
5098 { encoding = T5,
5099 <bch> family = bch,
5100 <cmr> family = cmr,
5101 shape = it }
5102 {
5103 <bch> _ = { ,100},
5104 <cmr> _ = {100,200},
5105 <bch> \textbackslash = {150,150},
5106 <cmr> \textbackslash = {300,300},
5107 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
5108 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5109 <bch> \guilsinglleft = {300,400}, \guilsingright = {200,500},
5110 <cmr> \guilsinglleft = {500,300}, \guilsingright = {400,400},
5111 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
5112 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
5113 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
5114 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
5115 <bch> \textless = {100, }, \textgreater = { ,100}
5116 <cmr> \textless = {300,100}, \textgreater = {200,100}
5117 }
5118
5119 </cmr|bch>

```

Slanted is very similar to italic.

```

5120 <*cmr>
5121 \SetProtrusion
5122 [ name = cmr-sl,
5123 load = cmr-it-OT1 ]
5124 { encoding = {OT1,OT4},
5125 family = cmr,
5126 shape = sl }
5127 {
5128 L = { ,50},
5129 f = { , -50},
5130 - = {300, },
5131 \textendash = {400, }, \textemdash = {300, }
5132 }
5133
5134 \SetProtrusion
5135 [ name = cmr-sl-T1,
5136 load = cmr-it-T1 ]
5137 { encoding = {T1,LY1},
5138 family = cmr,
5139 shape = sl }
5140 {
5141 L = { ,50},
5142 f = { , -50},
5143 - = {300, },

```

```

5144 \textendash = {400, }, \textendash = {300, }
5145 }
5146
5147 \SetProtrusion
5148 [ name = cmr-sl-T5,
5149   load = cmr-it-T5 ]
5150 { encoding = T5,
5151   family = cmr,
5152   shape = sl }
5153 {
5154   L = { ,50},
5155   f = { ,-50},
5156   - = {300, },
5157   \textendash = {400, }, \textendash = {300, }
5158 }
5159
5160 \SetProtrusion
5161 [ name = lmr-it-T1,
5162   load = cmr-it-T1 ]
5163 { encoding = {T1,LY1},
5164   family = lmr,
5165   shape = {it,sl} }
5166 {
5167   \textquotedblleft = {700,100},
5168   \quotedblbase = {600,300}
5169 }
5170

```

Oldstyle numerals are slightly different.

```

5171 \SetProtrusion
5172 [ name = cmr(oldstyle)-it,
5173   load = cmr-it-T1 ]
5174 { encoding = T1,
5175   family = {hfor,cmor},
5176   shape = {it,sl} }
5177 {
5178   1 = {250, 50},
5179   2 = {150,-100},
5180   3 = {100,-50},
5181   4 = {150,150},
5182   6 = {200, },
5183   7 = {200, 50},
5184   8 = {150,-50},
5185   9 = {100, 50}
5186 }
5187
5188 </cmr>
5189 <*pmn>
5190 \SetProtrusion
5191 [ name = pmnx-it,
5192   load = pmnj-it ]
5193 { encoding = OT1,
5194   family = pmnx,
5195   shape = {it,sl} }
5196 {
5197   1 = {100,150}
5198 }
5199
5200 \SetProtrusion
5201 [ name = pmnx-it-T1,
5202   load = pmnj-it-T1 ]
5203 { encoding = {T1,LY1},
5204   family = pmnx,

```

```

5205     shape    = {it,sl} }
5206   {
5207     1 = {100,150}
5208   }
5209
5210 </pmn>
5211 <*ptm>
5212 \SetProtrusion
5213   [ name    = ptm-it-LY1,
5214     load    = ptm-it-T1 ]
5215   { encoding = {LY1},
5216     family   = {ptm,ptmx,ptmj},
5217     shape    = {it,sl} }
5218   {
5219     -                               = {100,100},
5220     \texttrademark                 = {100,100},
5221     \textregistered                = {100,100},
5222     \textcopyright                 = {100,100},
5223     \textdegree                    = {300,100},
5224     \textminus                     = {200,200},
5225     \textellipsis                  = {100,200},
5226     \texteuro                      = { , },
5227     \textcent                     = {100,100},
5228     \textquotesingle               = {500, },
5229     \textflorin                    = {100, 70},
5230     \textdagger                    = {150,150},
5231     \textdaggerdbl                 = {100,100},
5232     \textbullet                    = {150,150},
5233     \textonesuperior               = {150,100},
5234     \texttwosuperior               = {150, 50},
5235     \textthreesuperior             = {150, 50},
5236     \textparagraph                 = {100, },
5237     \textperiodcentered             = {500,300},
5238     \textonequarter                = { 50, },
5239     \textonehalf                   = { 50, },
5240     \textplusminus                 = {100,100},
5241     \textmultiply                  = {150,150},
5242     \textdivide                    = {150,150}
5243   }
5244
5245 </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).

```

5246 <*(blg|ugm)>
5247 \SetProtrusion
5248 <m-t> [ name    = OT1-sc,
5249 <bch> [ name    = bch-sc,
5250 <cmr> [ name    = cmr-sc-OT1,
5251 <pad> [ name    = pad-sc,
5252 <pmn> [ name    = pmnj-sc,
5253 <ppl> [ name    = ppl-sc,
5254 <ptm> [ name    = ptm-sc,
5255 <m-t>   load    = default ]
5256 <bch>   load    = bch-default ]
5257 <cmr>   load    = cmr-OT1 ]
5258 <pad>   load    = pad-default ]
5259 <pmn>   load    = pmnj-default ]

```

```

5260 <ppl>      load      = ppl-default ]
5261 <ptm>      load      = ptm-default ]
5262 <m-t|bch|pad|pmn> { encoding = OT1,
5263 <cmr|ppl|ptm>    { encoding = {OT1,OT4},
5264 <bch>        family   = bch,
5265 <cmr>        family   = cmr,
5266 <pad>        family   = {pad,padx,padj},
5267 <pmn>        family   = pmnj,
5268 <ppl>        family   = {ppl,pplx,pplj},
5269 <ptm>        family   = {ptm,ptmx,ptmj},
5270      shape     = sc }
5271 {
5272   a = {50,50},
5273 <cmr|pad|ppl|ptm> \ae = {50, },
5274 <bch|pmn>         c = {50, },
5275 <bch|pad|pmn>     d = { ,50},
5276 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
5277 <bch|pad|pmn>     g = {50, },
5278 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
5279 <bch>             j = {100, },
5280 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
5281 <ptm>             l = { ,80},
5282 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % f1
5283 <ptm>             013 = { ,80}, % f1
5284 <bch|pad|pmn>     o = {50,50},
5285 <pad|pmn>         \oe = {50, },
5286 <ppl>             p = { 0, 0},
5287 <bch|pad|pmn>     q = {50,70},
5288 <ppl>             q = { 0, },
5289 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5290 t = {50,50},
5291 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5292 <ptm>            y = {80,80}
5293 }
5294
5295 \SetProtrusion
5296 <m-t> [ name      = T1-sc,
5297 <bch> [ name      = bch-sc-T1,
5298 <cmr> [ name      = cmr-sc-T1,
5299 <pad> [ name      = pad-sc-T1,
5300 <pmn> [ name      = pmnj-sc-T1,
5301 <ppl> [ name      = ppl-sc-T1,
5302 <ptm> [ name      = ptm-sc-T1,
5303 <m-t> load      = T1-default ]
5304 <bch> load      = bch-T1 ]
5305 <cmr> load      = cmr-T1 ]
5306 <pad> load      = pad-T1 ]
5307 <pmn> load      = pmnj-T1 ]
5308 <ppl> load      = ppl-T1 ]
5309 <ptm> load      = ptm-T1 ]
5310 { encoding = {T1,LY1},
5311 <bch>      family   = bch,
5312 <cmr>      family   = cmr,
5313 <pad>      family   = {pad,padx,padj},
5314 <pmn>      family   = pmnj,
5315 <ppl>      family   = {ppl,pplx,pplj},
5316 <ptm>      family   = {ptm,ptmx,ptmj},
5317   shape     = sc }
5318 {
5319   a = {50,50},
5320 <cmr|pad|ppl|ptm> \ae = {50, },
5321 <bch|pmn>         c = {50, },
5322 <bch|pad|pmn>     d = { ,50},

```

```

5323 <m-t|bch|cmr|pad|pmn|ptm>    f = { ,50},
5324 <bch|pad|pmn>    g = {50, },
5325 <m-t|cmr|pad|pmn|ppl|ptm>    j = {50, },
5326 <bch>    j = {100, },
5327 <m-t|bch|cmr|pad|pmn|ppl>    l = { ,50},
5328 <ptm>    l = { ,80},
5329 <m-t|bch|cmr|pad|pmn|ppl>    029 = { ,50}, % fl
5330 <ptm>    029 = { ,80}, % fl
5331 <bch|pad|pmn>    o = {50,50},
5332 <bch|pad|pmn>    \oe = {50, },
5333 <ppl>    p = { 0, 0},
5334 <bch|pad|pmn>    q = {50,70},
5335 <ppl>    q = { 0, },
5336 <m-t|cmr|pad|pmn|ppl|ptm>    r = { , 0},
5337    t = {50,50},
5338 <m-t|bch|cmr|pad|pmn|ppl>    y = {50,50}
5339 <ptm>    y = {80,80}
5340 }
5341
5342 </(big|ugm)>
5343 <*m-t>
5344 \SetProtrusion
5345 [ name      = QX-sc,
5346   load      = QX-default ]
5347 { encoding = QX,
5348   shape     = sc }
5349 {
5350   a = {50,50},
5351   f = { ,50},
5352   j = {50, },
5353   l = { ,50},
5354   013 = { ,50}, % fl
5355   r = { , 0},
5356   t = {50,50},
5357   y = {50,50}
5358 }
5359
5360 </m-t>
5361 <*cmr|bch>
5362 \SetProtrusion
5363 <bch> [ name      = bch-sc-T5,
5364 <bch>   load      = bch-T5 ]
5365 <cmr> [ name      = cmr-sc-T5,
5366 <cmr>   load      = cmr-T5 ]
5367 { encoding = T5,
5368 <bch>   family   = bch,
5369 <cmr>   family   = cmr,
5370   shape    = sc }
5371 {
5372   a = {50,50},
5373 <bch>   c = {50, },
5374 <bch>   d = { ,50},
5375   f = { ,50},
5376 <bch>   g = {50, },
5377 <bch>   j = {100, },
5378 <cmr>   j = {50, },
5379   l = { ,50},
5380 <bch>   o = {50,50},
5381 <bch>   q = { 0, },
5382 <cmr>   r = { , 0},
5383   t = {50,50},
5384   y = {50,50}
5385 }

```

```

5386
5387 </cmr>bch>
5388 <*pmn>
5389 \SetProtrusion
5390 [ name = pmnx-sc,
5391   load = pmnj-sc ]
5392 { encoding = OT1,
5393   family = pmnx,
5394   shape = sc }
5395 {
5396   1 = {230,180}
5397 }
5398
5399 \SetProtrusion
5400 [ name = pmnx-sc-T1,
5401   load = pmnj-sc-T1 ]
5402 { encoding = {T1,LY1},
5403   family = pmnx,
5404   shape = sc }
5405 {
5406   1 = {230,180}
5407 }
5408

```

#### 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`.

```

5409 \SetProtrusion
5410 [ name = pmnj-scit,
5411   load = pmnj-it ]
5412 { encoding = OT1,
5413   family = pmnj,
5414   shape = {scit,si} }
5415 {
5416   a = {50, },
5417   \ae = { , -50},
5418   b = {20, -50},
5419   c = {50, -50},
5420   d = {20, 0},
5421   e = {20, -50},
5422   f = {10, 0},
5423   012 = {10, -50}, % fi
5424   013 = {10, -50}, % fl
5425   014 = {10, -50}, % ffi
5426   015 = {10, -50}, % ffl
5427   g = {50, -50},
5428   i = {20, -50},
5429   j = {20, 0},
5430   k = {20, },
5431   l = {20, 50},
5432   m = { , -30},
5433   n = { , -30},
5434   o = {50, },
5435   \oe = {50, -50},
5436   p = {20, -50},
5437   q = {50, },
5438   r = {20, 0},
5439   s = {20, -30},
5440   t = {70, },
5441   u = {50, -50},

```



```

5442     v = {100, },
5443     w = {100, },
5444     y = {50, },
5445     z = { , -50}
5446 }
5447
5448 \SetProtrusion
5449 [ name      = pmnj-scit-T1,
5450   load      = pmnj-it-T1 ]
5451 { encoding = {T1,LY1},
5452   family   = pmnj,
5453   shape     = {scit,si} }
5454 {
5455   a = {50, },
5456   \ae = { , -50},
5457   b = {20, -50},
5458   c = {50, -50},
5459   d = {20, 0},
5460   e = {20, -50},
5461   f = {10, 0},
5462   028 = {10, -50}, % fi
5463   029 = {10, -50}, % fl
5464   030 = {10, -50}, % ffi
5465   031 = {10, -50}, % ffl
5466   g = {50, -50},
5467   i = {20, -50},
5468   188 = {20, 0}, % ij
5469   j = {20, 0},
5470   k = {20, },
5471   l = {20, 50},
5472   m = { , -30},
5473   n = { , -30},
5474   o = {50, },
5475   \oe = {50, -50},
5476   p = {20, -50},
5477   q = {50, },
5478   r = {20, 0},
5479   s = {20, -30},
5480   t = {70, },
5481   u = {50, -50},
5482   v = {100, },
5483   w = {100, },
5484   y = {50, },
5485   z = { , -50}
5486 }
5487
5488 \SetProtrusion
5489 [ name      = pmnx-scit,
5490   load      = pmnj-scit ]
5491 { encoding = OT1,
5492   family   = pmnx,
5493   shape     = {scit,si} }
5494 {
5495   1 = {100,150}
5496 }
5497
5498 \SetProtrusion
5499 [ name      = pmnx-scit-T1,
5500   load      = pmnj-scit-T1 ]
5501 { encoding = {T1,LY1},
5502   family   = pmnx,
5503   shape     = {scit,si} }
5504 {

```

```

5505     1 = {100,150}
5506   }
5507
5508 </pmn>

```

### 15.8.5 textcomp

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

5509 \SetProtrusion
5510 <m-t> [ name      = textcomp ]
5511 <bch> [ name      = bch-textcomp ]
5512 <blg> [ name      = blg-textcomp ]
5513 <cmr> [ name      = cmr-textcomp ]
5514 <pad> [ name      = pad-textcomp ]
5515 <pmn> [ name      = pmn-textcomp ]
5516 <ppl> [ name      = ppl-textcomp ]
5517 <ptm> [ name      = ptm-textcomp ]
5518 <ugm> [ name      = ugm-textcomp ]
5519 <m-t> { encoding = TS1 }
5520 <!m-t> { encoding = TS1,
5521       <bch> family = bch }
5522       <blg> family = blg }
5523       <cmr> family = cmr }
5524       <pad> family = {pad,padx,padj} }
5525       <pmn> family = {pmnx,pmnj} }
5526       <ppl> family = {ppl,pplx,pplj} }
5527       <ptm> family = {ptm,ptmx,ptmj} }
5528       <ugm> family = ugm }
5529 {
5530   <blg> \textquotestraightbase = {400,500},
5531   <cmr> \textquotestraightbase = {300,300},
5532   <pad|pmn> \textquotestraightbase = {400,400},
5533   <blg> \textquotestraightdblbase = {300,400},
5534   <cmr|pmn> \textquotestraightdblbase = {300,300},
5535   <pad> \textquotestraightdblbase = {400,400},
5536   <bch|cmr|pad|pmn|ugm> \texttwelveudash = {200,200},
5537   <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
5538   <ugm> \textthreequartersemdash = {200,200},
5539   <blg> \textquotesingle = {500,600},
5540   <cmr|pmn> \textquotesingle = {300,400},
5541   <pad> \textquotesingle = {400,500},
5542   <ptm> \textquotesingle = {500,500},
5543   <ugm> \textquotesingle = {300,500},
5544   <bch|cmr|pmn> \textasteriskcentered = {200,300},
5545   <blg> \textasteriskcentered = {150,200},
5546   <pad> \textasteriskcentered = {300,300},
5547   <ugm> \textasteriskcentered = {100,200},
5548   <pmn> \textfractionsolidus = {-200,-200},
5549   <cmr> \textoneoldstyle = {100,100},
5550   <pmn> \textoneoldstyle = { , 50},
5551   <cmr> \textthreeoldstyle = { , 50},
5552   <pad|pmn> \textthreeoldstyle = { 50, },
5553   <cmr> \textfouroldstyle = { 50, 50},
5554   <pad|pmn> \textfouroldstyle = { 50, },
5555   <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
5556   <cmr> \textlangle = {400, },
5557   <cmr> \textrangle = { ,400},
5558   <m-t|bch|pmn|ptm> \textminus = {200,200},
5559   <cmr|pad|ppl> \textminus = {300,300},
5560   <blg|ugm> \textminus = {250,300},

```

```

5561 <bch|pad|pmn> \textlbrackdbl = {100, },
5562 <blg> \textlbrackdbl = {200, },
5563 <bch|pad|pmn> \textrbrackdbl = { ,100},
5564 <blg> \textrbrackdbl = { ,200},
5565 <pmn> \textasciigrave = {200,500},
5566 <bch|blg|cmr|pad|pmn> \texttildebelow = {200,250},
5567 <pmn> \textasciibreve = {300,400},
5568 <pmn> \textasciicaron = {300,400},
5569 <pmn> \textacutedbl = {200,300},
5570 <pmn> \textgravedbl = {150,300},
5571 <bch|pmn|ugm> \textdagger = { 80, 80},
5572 <blg> \textdagger = {200,200},
5573 <cmr|pad> \textdagger = {100,100},
5574 <ptm> \textdagger = {150,150},
5575 <blg> \textdaggerdbl = {150,150},
5576 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
5577 <ptm> \textdaggerdbl = {100,100},
5578 <bch> \textbardbl = {100,100},
5579 <blg|ugm> \textbardbl = {150,150},
5580 <bch> \textbullet = {200,200},
5581 <blg> \textbullet = {400,500},
5582 <cmr|pad|pmn> \textbullet = { ,100},
5583 <ptm> \textbullet = {150,150},
5584 <ugm> \textbullet = { 50,100},
5585 <bch|cmr|pmn> \textcelsius = { 50, },
5586 <pad> \textcelsius = { 80, },
5587 <bch> \textflorin = { 50, 50},
5588 <blg> \textflorin = {100,100},
5589 <pad|ugm> \textflorin = { ,100},
5590 <pmn> \textflorin = { 50,100},
5591 <ptm> \textflorin = { 50, 70},
5592 <cmr> \textcolonmonetary = { , 50},
5593 <pad|pmn> \textcolonmonetary = { 50, },
5594 <pmn> \textinterrobang = { ,100},
5595 <pmn> \textinterrobangdown = {100, },
5596 <m-t|pad|ptm> \texttrademark = {100,100},
5597 <bch> \texttrademark = {150,150},
5598 <blg|cmr|ppl> \texttrademark = {200,200},
5599 <pmn> \texttrademark = { 50, 50},
5600 <ugm> \texttrademark = {100,150},
5601 <bch|ugm> \textcent = { 50, },
5602 <ptm> \textcent = {100,100},
5603 <bch> \textsterling = { 50, },
5604 <ugm> \textsterling = { , 50},
5605 <bch> \textbrokenbar = {200,200},
5606 <blg> \textbrokenbar = {250,250},
5607 <ugm> \textbrokenbar = {200,300},
5608 <pmn> \textasciidieresis = {300,400},
5609 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
5610 <pmn> \textcopyright = {100,150},
5611 <ppl> \textcopyright = {200,200},
5612 <bch|cmr|ugm> \textordfeminine = {100,200},
5613 <pad|pmn> \textordfeminine = {200,200},
5614 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
5615 <blg> \textlnot = {200,100},
5616 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
5617 <pmn> \textregistered = { 50,150},
5618 <ppl> \textregistered = {200,200},
5619 <pmn> \textasciimacron = {150,200},
5620 <m-t|ppl|ptm> \textdegree = {300,300},
5621 <bch> \textdegree = {150,200},
5622 <blg|ugm> \textdegree = {200,200},
5623 <cmr|pad> \textdegree = {400,400},

```

```

5624 <pmn> \textdegree = {150,400},
5625 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
5626 <blg> \textpm = {100,100},
5627 <ptm> \textpm = { 50, 80},
5628 <bch|blg|ugm> \texttwosuperior = {100,200},
5629 <cmr> \texttwosuperior = { 50,100},
5630 <pad|pmn> \texttwosuperior = {200,200},
5631 <ptm> \texttwosuperior = { 50, 50},
5632 <bch|blg|ugm> \textthreesuperior = {100,200},
5633 <cmr> \textthreesuperior = { 50,100},
5634 <pad|pmn> \textthreesuperior = {200,200},
5635 <ptm> \textthreesuperior = { 50, 50},
5636 <pmn> \textasciicute = {300,400},
5637 <bch|ugm> \textmu = { ,100},
5638 <bch|pad|pmn> \textparagraph = { ,100},
5639 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
5640 <blg> \textperiodcentered = {400,500},
5641 <ptm> \textperiodcentered = {300,300},
5642 <ugm> \textperiodcentered = {200,500},
5643 <bch|blg|ugm> \textonesuperior = {200,300},
5644 <cmr|pad|pmn> \textonesuperior = {200,200},
5645 <ptm> \textonesuperior = {100,100},
5646 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
5647 <blg|cmr> \textordmasculine = {100,200},
5648 <bch|cmr|pmn> \texteuro = {100, },
5649 <pad> \texteuro = { 50,100},
5650 <bch> \texttimes = {200,200},
5651 <blg|ptm> \texttimes = {100,100},
5652 <cmr> \texttimes = {150,250},
5653 <pad> \texttimes = {100,150},
5654 <pmn> \texttimes = { 70,100},
5655 <ugm> \texttimes = {200,300},
5656 <bch|pad|pmn> \textdiv = {150,200}
5657 <blg> \textdiv = {100,100}
5658 <cmr> \textdiv = {150,250}
5659 <ptm> \textdiv = { 50,100},
5660 <ugm> \textdiv = {200,300},
5661 <ptm> \textperthousand = { ,50}
5662 <ugm> \textsection = { ,100},
5663 <ugm> \textonehalf = { 50,100},
5664 <ugm> \textonequarter = { 50,100},
5665 <ugm> \textthreequarters = { 50,100},
5666 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

5667 }
5668
5669 <*cmr|pad|pmn|ugm>
5670 \SetProtrusion
5671 <cmr> [ name = cmr-textcomp-it ]
5672 <pad> [ name = pad-textcomp-it ]
5673 <pmn> [ name = pmn-textcomp-it ]
5674 <ugm> [ name = ugm-textcomp-it ]
5675 { encoding = TS1,
5676 <cmr> family = cmr,
5677 <pad> family = {pad,padx,padj},
5678 <pmn> family = {pmnx,pmnj},
5679 <ugm> family = ugm,
5680 <!ugm> shape = {it,s1} }
5681 <ugm> shape = it }
5682 {
5683 <cmr> \textquotestraightbase = {300,600},
5684 <pad|pmn> \textquotestraightbase = {400,400},

```

```

5685 <cmr> \textquotestraightdblbase = {300,600},
5686 <pad> \textquotestraightdblbase = {300,400},
5687 <pmn> \textquotestraightdblbase = {300,300},
5688 \texttwelvewardash = {200,200},
5689 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
5690 <ugm> \textthreequartersemdash = {200,200},
5691 <cmr> \textquotesingle = {600,300},
5692 <pad> \textquotesingle = {800,100},
5693 <pmn> \textquotesingle = {300,200},
5694 <ugm> \textquotesingle = {500,500},
5695 <cmr> \textasteriskcentered = {300,200},
5696 <pad> \textasteriskcentered = {500,100},
5697 <pmn> \textasteriskcentered = {200,300},
5698 <ugm> \textasteriskcentered = {300,150},
5699 <pmn> \textfactionsolidus = {-200,-200},
5700 <cmr> \textoneoldstyle = {100, 50},
5701 <pad> \textoneoldstyle = {100, },
5702 <pmn> \textoneoldstyle = { 50, },
5703 <pad> \texttwooldstyle = { 50, },
5704 <pmn> \texttwooldstyle = {-50, },
5705 <cmr> \textthreeoldstyle = {100, 50},
5706 <pmn> \textthreeoldstyle = {-100, },
5707 <cmr> \textfouroldstyle = { 50, 50},
5708 <pad> \textfouroldstyle = { 50,100},
5709 <cmr> \textsevenoldstyle = { 50, 80},
5710 <pad> \textsevenoldstyle = { 50, },
5711 <pmn> \textsevenoldstyle = { 20, },
5712 <cmr> \textlangle = {400, },
5713 <cmr> \textrangle = { ,400},
5714 <cmr|pad> \textminus = {300,300},
5715 <pmn> \textminus = {200,200},
5716 <ugm> \textminus = {250,300},
5717 <pad|pmn> \textlbrackdbl = {100, },
5718 <pad|pmn> \textrbrackdbl = { ,100},
5719 <pmn> \textasciigrave = {300,300},
5720 <cmr|pad|pmn> \texttildelow = {200,250},
5721 <pmn> \textasciibreve = {300,300},
5722 <pmn> \textasciicaron = {300,300},
5723 <pmn> \textacutedbl = {200,300},
5724 <pmn> \textgravedbl = {150,300},
5725 <cmr> \textdagger = {100,100},
5726 <pad> \textdagger = {200,100},
5727 <pmn> \textdagger = { 80, 50},
5728 <ugm> \textdagger = { 80, 80},
5729 <cmr|pad> \textdaggerdbl = { 80, 80},
5730 <pmn> \textdaggerdbl = { 80, 50},
5731 <ugm> \textbardbl = {150,150},
5732 <cmr> \textbullet = {200,100},
5733 <pad> \textbullet = {300, },
5734 <pmn> \textbullet = { 30, 70},
5735 <ugm> \textbullet = { 50,100},
5736 <cmr> \textcelsius = {100, },
5737 <pad> \textcelsius = {200, },
5738 <pmn> \textcelsius = { 50,-50},
5739 <pad> \textflorin = {100, },
5740 <pmn> \textflorin = { 50,100},
5741 <ugm> \textflorin = { ,100},
5742 <cmr> \textcolonmonetary = {150, },
5743 <pad> \textcolonmonetary = {100, },
5744 <pmn> \textcolonmonetary = { 50,-50},
5745 <cmr|pad> \texttrademark = {200, },
5746 <pmn> \texttrademark = { 50,100},
5747 <ugm> \texttrademark = {150, 50},

```

```

5748 <ugm> \textcent = { 50, },
5749 <ugm> \textsterling = { , 50},
5750 <ugm> \textbrokenbar = {200,300},
5751 <pmn> \textasciidieresis = {300,200},
5752 <cmr> \textcopyright = {100, },
5753 <pad> \textcopyright = {200,100},
5754 <pmn> \textcopyright = {100,150},
5755 <ugm> \textcopyright = {300, },
5756 <cmr> \textordfeminine = {100,100},
5757 <pmn> \textordfeminine = {200,200},
5758 <ugm> \textordfeminine = {100,200},
5759 <cmr|pad> \textlnot = {300, },
5760 <pmn|ugm> \textlnot = {200, },
5761 <cmr> \textregistered = {100, },
5762 <pad> \textregistered = {200,100},
5763 <pmn> \textregistered = { 50,150},
5764 <ugm> \textregistered = {300, },
5765 <pmn> \textasciimacron = {150,200},
5766 <cmr|pad> \textdegree = {500,100},
5767 <pmn> \textdegree = {150,150},
5768 <ugm> \textdegree = {300,200},
5769 <cmr> \textpm = {150,100},
5770 <pad> \textpm = {200,150},
5771 <pmn|ugm> \textpm = {150,200},
5772 <cmr> \textonesuperior = {400, },
5773 <pad> \textonesuperior = {300,100},
5774 <pmn> \textonesuperior = {200,100},
5775 <ugm> \textonesuperior = {300,300},
5776 <cmr> \texttwosuperior = {400, },
5777 <pad> \texttwosuperior = {300, },
5778 <pmn> \texttwosuperior = {200,100},
5779 <ugm> \texttwosuperior = {300,200},
5780 <cmr> \textthreesuperior = {400, },
5781 <pad> \textthreesuperior = {300, },
5782 <pmn> \textthreesuperior = {200,100},
5783 <ugm> \textthreesuperior = {300,200},
5784 <ugm> \textmu = { ,100},
5785 <pmn> \textasciicute = {300,200},
5786 <cmr> \textparagraph = {200, },
5787 <pmn> \textparagraph = { ,100},
5788 <cmr> \textperiodcentered = {500,500},
5789 <pad|pmn|ugm> \textperiodcentered = {300,400},
5790 <cmr> \textordmasculine = {100,100},
5791 <pmn> \textordmasculine = {200,200},
5792 <ugm> \textordmasculine = {300,200},
5793 <cmr> \texteuro = {200, },
5794 <pad> \texteuro = {100, },
5795 <pmn> \texteuro = {100,-50},
5796 <cmr> \texttimes = {200,200},
5797 <pad> \texttimes = {200,100},
5798 <pmn> \texttimes = { 70,100},
5799 <ugm> \texttimes = {200,300},
5800 <cmr|pad> \textdiv = {200,200}
5801 <pmn> \textdiv = {150,200}
5802 <ugm> \textdiv = {200,300},
5803 <ugm> \textsection = { ,200},
5804 <ugm> \textonehalf = { 50,100},
5805 <ugm> \textonequarter = { 50,100},
5806 <ugm> \textthreequarters = { 50,100},
5807 <ugm> \textsurd = { ,100}
5808 }
5809
5810 </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}
```

```
5811 <*cmr>
5812 \SetProtrusion
5813 [ name = cmr-math-letters ]
5814 { encoding = OML,
5815   family = cmm,
5816   series = {m,b},
5817   shape = it }
5818 {
5819   A = {100, 50}, % \mathnormal
5820   B = { 50,   },
5821   C = { 50,   },
5822   D = { 50, 50},
5823   E = { 50,   },
5824   F = {100, 50},
5825   G = { 50, 50},
5826   H = { 50, 50},
5827   I = { 50, 50},
5828   J = {150, 50},
5829   K = { 50,100},
5830   L = { 50, 50},
5831   M = { 50,   },
5832   N = { 50,   },
5833   O = { 50,   },
5834   P = { 50,   },
5835   Q = { 50, 50},
5836   R = { 50,   },
5837   S = { 50,   },
5838   T = { 50,100},
5839   U = { 50, 50},
5840   V = {100,100},
5841   W = { 50,100},
5842   X = { 50,100},
5843   Y = {100,100},
5844   f = {100,100},
5845   h = {   ,100},
5846   i = {   , 50},
5847   j = {   , 50},
5848   k = {   , 50},
5849   r = {   , 50},
5850   v = {   , 50},
5851   w = {   , 50},
5852   x = {   , 50},
5853   "OB = { 50,100}, % \alpha
```

```

5854 "0C = { 50, 50}, % \beta
5855 "0D = {200,150}, % \gamma
5856 "0E = { 50, 50}, % \delta
5857 "0F = { 50, 50}, % \epsilon
5858 "10 = { 50,150}, % \zeta
5859 "12 = { 50, }, % \theta
5860 "13 = { ,100}, % \iota
5861 "14 = { ,100}, % \kappa
5862 "15 = {100, 50}, % \lambda
5863 "16 = { , 50}, % \mu
5864 "17 = { , 50}, % \nu
5865 "18 = { , 50}, % \xi
5866 "19 = { 50,100}, % \pi
5867 "1A = { 50, 50}, % \rho
5868 "1B = { ,150}, % \sigma
5869 "1C = { 50,150}, % \tau
5870 "1D = { 50, 50}, % \upsilon
5871 "1F = { 50,100}, % \chi
5872 "20 = { 50, 50}, % \psi
5873 "21 = { , 50}, % \omega
5874 "22 = { , 50}, % \varepsilon
5875 "23 = { , 50}, % \vartheta
5876 "24 = { , 50}, % \varpi
5877 "25 = {100, }, % \varrho
5878 "26 = {100,100}, % \varsigma
5879 "27 = { 50, 50}, % \varphi
5880 "28 = {100,100}, % \leftharpoonup
5881 "29 = {100,100}, % \leftharpoondown
5882 "2A = {100,100}, % \rightharpoonup
5883 "2B = {100,100}, % \rightharpoondown
5884 "2C = {300,200}, % \lhook
5885 "2D = {200,300}, % \rhook
5886 "2E = { ,100}, % \triangleright
5887 "2F = {100, }, % \triangleleft
5888 "3A = { ,500}, % ., \ldotp
5889 "3B = { ,500}, % ,
5890 "3C = {200,100}, % <
5891 "3D = {300,400}, % /
5892 "3E = {100,200}, % >
5893 "3F = {200,200}, % \star
5894 "5B = { ,100}, % \flat
5895 "5E = {200,200}, % \smile
5896 "5F = {200,200}, % \frown
5897 "7C = {100, }, % \jmath
5898 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

5899 }
5900

```

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}

```

```

5901 \SetProtrusion
5902 [ name = cmr-math-symbols ]
5903 { encoding = OMS,
5904   family = cmsy,
5905   series = {m,b},
5906   shape = n }
5907 {
5908   A = {150, 50}, % \mathcal

```



```

5909     C = {    ,100},
5910     D = {    , 50},
5911     F = { 50,150},
5912     I = {    ,100},
5913     J = {100,150},
5914     K = {    ,100},
5915     L = {100,   },
5916     M = { 50, 50},
5917     N = { 50,100},
5918     P = {    , 50},
5919     Q = { 50,   },
5920     R = {    , 50},
5921     T = { 50,150},
5922     V = { 50, 50},
5923     W = {    , 50},
5924     X = {100,100},
5925     Y = {100,   },
5926     Z = {100,150},
5927     "00 = {300,300}, % -
5928     "01 = {    ,700}, % \cdot, \cdotp
5929     "02 = {150,250}, % \times
5930     "03 = {150,250}, % *, \ast
5931     "04 = {200,300}, % \div
5932     "05 = {150,250}, % \diamond
5933     "06 = {200,200}, % \pm
5934     "07 = {200,200}, % \mp
5935     "08 = {100,100}, % \oplus
5936     "09 = {100,100}, % \ominus
5937     "0A = {100,100}, % \otimes
5938     "0B = {100,100}, % \oslash
5939     "0C = {100,100}, % \odot
5940     "0D = {100,100}, % \bigcirc
5941     "0E = {100,100}, % \circ
5942     "0F = {100,100}, % \bullet
5943     "10 = {100,100}, % \asymp
5944     "11 = {100,100}, % \equiv
5945     "12 = {200,100}, % \subteq
5946     "13 = {100,200}, % \supseteq
5947     "14 = {200,100}, % \leq
5948     "15 = {100,200}, % \geq
5949     "16 = {200,100}, % \preceq
5950     "17 = {100,200}, % \succeq
5951     "18 = {200,200}, % \sim
5952     "19 = {150,150}, % \approx
5953     "1A = {200,100}, % \subset
5954     "1B = {100,200}, % \supset
5955     "1C = {200,100}, % \ll
5956     "1D = {100,200}, % \gg
5957     "1E = {300,100}, % \prec
5958     "1F = {100,300}, % \succ
5959     "20 = {100,200}, % \leftarrow
5960     "21 = {200,100}, % \rightarrow
5961     "22 = {100,100}, % \uparrow
5962     "23 = {100,100}, % \downarrow
5963     "24 = {100,100}, % \leftrightarrow
5964     "25 = {100,100}, % \nearrow
5965     "26 = {100,100}, % \searrow
5966     "27 = {100,100}, % \simeq
5967     "28 = {100,100}, % \Leftarrow
5968     "29 = {100,100}, % \Rightarrow
5969     "2A = {100,100}, % \Uparrow
5970     "2B = {100,100}, % \Downarrow
5971     "2C = {100,100}, % \Leftrightarrow

```

```

5972 "2D = {100,100}, % \narrow
5973 "2E = {100,100}, % \swarrow
5974 "2F = { ,100}, % \propto
5975 "30 = { ,400}, % \prime
5976 "31 = {100,100}, % \infty
5977 "32 = {150,100}, % \in
5978 "33 = {100,150}, % \ni
5979 "34 = {100,100}, % \triangle, \bigtriangleup
5980 "35 = {100,100}, % \bigtriangledown
5981 "38 = { ,100}, % \forall
5982 "39 = {100, }, % \exists
5983 "3A = {200, }, % \neg
5984 "3E = {200,200}, % \top
5985 "3F = {200,200}, % \bot, \perp
5986 "5E = {100,200}, % \wedge
5987 "5F = {100,200}, % \vee
5988 "60 = { ,300}, % \vdash
5989 "61 = {300, }, % \dashv
5990 "62 = {100,100}, % \lfloor
5991 "63 = {100,100}, % \rfloor
5992 "64 = {100,100}, % \lceil
5993 "65 = {100,100}, % \rceil
5994 "66 = {150, }, % \lbrace
5995 "67 = { ,150}, % \rbrace
5996 "68 = {400, }, % \langle
5997 "69 = { ,400}, % \rangle
5998 "6C = {100,100}, % \updownarrow
5999 "6D = {100,100}, % \Updownarrow
6000 "6E = {100,300}, % \, \backslash, \setminus
6001 "72 = {100,100}, % \nabla
6002 "79 = {200,200}, % \dagger
6003 "7A = {100,100}, % \ddagger
6004 "7B = {100, }, % \mathparagraph
6005 "7C = {100,100}, % \clubsuit
6006 "7D = {100,100}, % \diamondsuit
6007 "7E = {100,100}, % \heartsuit
6008 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

6009 }
6010

```

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}
```

```

6011 </cmr>
6012 </cfg-t>

```

### 15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
6013 <*cfg-u>
```

Symbol font 'a'.

```

6014 <*msa>
6015 \SetProtrusion
6016 [ name = AMS-a ]
6017 { encoding = U,
6018   family = msa }

```

```

6019 {
6020   "05 = {150,250}, % \centerdot
6021   "06 = {100,100}, % \lozenge
6022   "07 = { 50, 50}, % \blacklozenge
6023   "08 = { 50, 50}, % \circlearrowright
6024   "09 = { 50, 50}, % \circlearrowleft
6025   "0A = {100,100}, % \rightleftharpoons
6026   "0B = {100,100}, % \leftrightharpoons
6027   "0D = {-50,200}, % \Vdash
6028   "0E = {-50,200}, % \Vvdash
6029   "0F = {-70,150}, % \vDash
6030   "10 = {100,150}, % \twoheadrightarrow
6031   "11 = {100,150}, % \twoheadleftarrow
6032   "12 = { 50,100}, % \leftleftarrows
6033   "13 = { 50, 80}, % \rightrightarrows
6034   "14 = {120,120}, % \upuparrows
6035   "15 = {120,120}, % \downdownarrows
6036   "16 = {200,200}, % \upharpoonright
6037   "17 = {200,200}, % \downharpoonright
6038   "18 = {200,200}, % \upharpoonleft
6039   "19 = {200,200}, % \downharpoonleft
6040   "1A = { 80,100}, % \rightarrowtail
6041   "1B = { 80,100}, % \leftarrowtail
6042   "1C = { 50, 50}, % \leftrightarrows
6043   "1D = { 50, 50}, % \rightleftarrows
6044   "1E = {250,   }, % \Lsh
6045   "1F = {   ,250}, % \Rsh
6046   "20 = {100,100}, % \rightsquigarrow
6047   "21 = {100,100}, % \leftrightsquigarrow
6048   "22 = {100, 50}, % \looparrowleft
6049   "23 = { 50,100}, % \looparrowright
6050   "24 = { 50, 80}, % \circeq
6051   "25 = {   ,100}, % \succsim
6052   "26 = {   ,100}, % \gtrsim
6053   "27 = {   ,100}, % \gtrapprox
6054   "28 = {150, 50}, % \multimap
6055   "2B = {100,150}, % \doteqdot
6056   "2C = {100,150}, % \triangleq
6057   "2D = {100, 50}, % \precsim
6058   "2E = {100, 50}, % \lesssim
6059   "2F = { 50, 50}, % \lessapprox
6060   "30 = {100, 50}, % \eqslantless
6061   "31 = { 50, 50}, % \eqslantgtr
6062   "32 = {100, 50}, % \curlyeqprec
6063   "33 = { 50,100}, % \curlyeqsucc
6064   "34 = {100, 50}, % \preccurlyeq
6065   "36 = { 50,   }, % \leqslant
6066   "38 = {   , 50}, % \backprime
6067   "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
6068   "3C = { 50,100}, % \succcurlyeq
6069   "3E = {   , 50}, % \geqslant
6070   "40 = {   , 50}, % \sqsubset
6071   "41 = { 50,   }, % \sqsupset
6072   "42 = {   ,150}, % \vartriangleright, \rhd
6073   "43 = {150,   }, % \vartriangleleft, \lhd
6074   "44 = {   ,100}, % \trianglerighteq, \unrhd
6075   "45 = {100,   }, % \trianglelefteq, \unlhd
6076   "46 = {100,100}, % \bigstar
6077   "48 = { 50, 50}, % \blacktriangledown
6078   "49 = {   ,100}, % \blacktriangleright
6079   "4A = {100,   }, % \blacktriangleleft
6080   "4B = {   ,150}, % \dashrightarrow (the arrow)
6081   "4C = {150,   }, % \dashleftarrow

```

```

6082 "4D = { 50, 50}, % \vartriangle
6083 "4E = { 50, 50}, % \blacktriangle
6084 "4F = { 50, 50}, % \triangledown
6085 "50 = { 50, 50}, % \eqcirc
6086 "56 = { ,150}, % \Rrightarrow
6087 "57 = {150, }, % \Lleftarrow
6088 "58 = {100,300}, % \checkmark
6089 "5C = { 50, 50}, % \angle
6090 "5D = { 50, 50}, % \measuredangle
6091 "5E = { 50, 50}, % \sphericalangle
6092 "5F = { , 50}, % \varpropto
6093 "60 = {100,100}, % \smallsmile
6094 "61 = {100,100}, % \smallfrown
6095 "62 = { 50, }, % \Subset
6096 "63 = { , 50}, % \Supset
6097 "66 = {150,150}, % \curlywedge
6098 "67 = {150,150}, % \curlyvee
6099 "68 = { 50,150}, % \leftthreetimes
6100 "69 = {100, 50}, % \rightthreetimes
6101 "6C = { 50, 50}, % \bumpeq
6102 "6D = { 50, 50}, % \Bumpeq
6103 "6E = {100, }, % \lll
6104 "6F = { ,100}, % \ggg
6105 "70 = { 50,100}, % \ulcorner
6106 "71 = {100, 50}, % \urcorner
6107 "75 = {150,200}, % \dotplus
6108 "76 = { 50,100}, % \backsim
6109 "78 = { 50,100}, % \llcorner
6110 "79 = {100, 50}, % \lrcorner
6111 "7C = {100,100}, % \intercal
6112 "7D = { 50, 50}, % \circledcirc
6113 "7E = { 50, 50}, % \circledast
6114 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

6115 }
6116
6117 </msa>

```

Symbol font 'b'.

```

6118 <*msb>
6119 \SetProtrusion
6120 [ name = AMS-b ]
6121 { encoding = U,
6122   family = msb }
6123 {
6124   A = { 50, 50}, % \mathbb
6125   C = { 50, 50},
6126   G = { , 50},
6127   L = { , 50},
6128   P = { , 50},
6129   R = { , 50},
6130   T = { , 50},
6131   V = { 50, 50},
6132   X = { 50, 50},
6133   Y = { 50, 50},
6134   "00 = { 50, 50}, % \lvertneqq
6135   "01 = { 50, 50}, % \gvertneqq
6136   "02 = { 50, 50}, % \nleq
6137   "03 = { 50, 50}, % \ngeq
6138   "04 = {100, 50}, % \nless
6139   "05 = { 50,150}, % \ngtr
6140   "06 = {100, 50}, % \nprec

```

```

6141 "07 = { 50,150}, % \nsucc
6142 "08 = { 50, 50}, % \lneqq
6143 "09 = { 50, 50}, % \gneqq
6144 "0A = {100,100}, % \lneqslant
6145 "0B = {100,100}, % \ngeqslant
6146 "0C = {100, 50}, % \lneq
6147 "0D = { 50,100}, % \gneq
6148 "0E = {100, 50}, % \npreceq
6149 "0F = { 50,100}, % \nsucceq
6150 "10 = { 50,  }, % \precnsim
6151 "11 = { 50, 50}, % \succnsim
6152 "12 = { 50, 50}, % \lnsim
6153 "13 = { 50, 50}, % \gnsim
6154 "14 = { 50, 50}, % \lneqq
6155 "15 = { 50, 50}, % \ngeqq
6156 "16 = { 50, 50}, % \precneqq
6157 "17 = { 50, 50}, % \succneqq
6158 "18 = { 50, 50}, % \precnapprox
6159 "19 = { 50, 50}, % \succnapprox
6160 "1A = { 50, 50}, % \lnapprox
6161 "1B = { 50, 50}, % \gnapprox
6162 "1C = {150,200}, % \nsim
6163 "1D = { 50, 50}, % \ncong
6164 "1E = {100,150}, % \diagup
6165 "1F = {100,150}, % \diagdown
6166 "20 = {100, 50}, % \varsubsetneq
6167 "21 = { 50,100}, % \varsupsetneq
6168 "22 = {100, 50}, % \subsetneqq
6169 "23 = { 50,100}, % \supsetneqq
6170 "24 = {100, 50}, % \subsetneqq
6171 "25 = { 50,100}, % \supsetneqq
6172 "26 = {100, 50}, % \varsubsetneqq
6173 "27 = { 50,100}, % \varsupsetneqq
6174 "28 = {100, 50}, % \subsetneq
6175 "29 = { 50,100}, % \supsetneq
6176 "2A = {100, 50}, % \subseteq
6177 "2B = { 50,100}, % \supseteq
6178 "2C = { 50,100}, % \nparallel
6179 "2D = {100,150}, % \nmid
6180 "2E = {150,150}, % \nshortmid
6181 "2F = {100,100}, % \nshortparallel
6182 "30 = {  ,150}, % \nvdash
6183 "31 = {  ,150}, % \nVdash
6184 "32 = {  ,100}, % \nvDash
6185 "33 = {  ,100}, % \nVDash
6186 "34 = {  ,100}, % \ntrianglerighteq
6187 "35 = {100,  }, % \trianglelefteq
6188 "36 = {100,  }, % \triangleleft
6189 "37 = {  ,100}, % \triangleright
6190 "38 = {100,200}, % \leftarrow
6191 "39 = {100,200}, % \rightarrow
6192 "3A = {100,100}, % \Leftarrow
6193 "3B = { 50,100}, % \Rightarrow
6194 "3C = {100,100}, % \Leftrightarrow
6195 "3D = {100,200}, % \leftrightarrows
6196 "3E = { 50, 50}, % \divideontimes
6197 "3F = { 50, 50}, % \varnothing
6198 "60 = {200,  }, % \Finv
6199 "61 = {  , 50}, % \Game
6200 "68 = {100,100}, % \eqsim
6201 "69 = { 50,  }, % \beth
6202 "6A = { 50,  }, % \gimel
6203 "6B = {150,  }, % \daleth

```

```

6204 "6C = {200, }, % \lessdot
6205 "6D = { ,200}, % \gtrdot
6206 "6E = {100,200}, % \ltimes
6207 "6F = {150,100}, % \rtimes
6208 "70 = { 50,100}, % \shortmid
6209 "71 = { 50, 50}, % \shortparallel
6210 "72 = {200,300}, % \smallsetminus
6211 "73 = {100,200}, % \thicksim
6212 "74 = { 50,100}, % \thickapprox
6213 "75 = { 50, 50}, % \approxeq
6214 "76 = { 50,100}, % \succapprox
6215 "77 = { 50, 50}, % \precapprox
6216 "78 = {100,100}, % \curvearrowleft
6217 "79 = { 50,150}, % \curvearrowright
6218 "7A = { 50,200}, % \digamma
6219 "7B = {100, 50}, % \varkappa
6220 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

6221 }
6222
6223 </msb>

```

### 15.8.8 Euler

Euler Roman font (package `euler`).

```

6224 <*eur>
6225 \SetProtrusion
6226 [ name = euler ]
6227 { encoding = U,
6228   family = eur }
6229 {
6230   "01 = {100,100},
6231   "03 = {100,150},
6232   "06 = { ,100},
6233   "07 = {100,150},
6234   "08 = {100,100},
6235   "0A = {100,100},
6236   "0B = { , 50},
6237   "0C = { ,100},
6238   "0D = {100,100},
6239   "0E = { ,100},
6240   "0F = {100,100},
6241   "10 = {100,100},
6242   "13 = { ,100},
6243   "14 = { ,100},
6244   "15 = { , 50},
6245   "16 = { , 50},
6246   "17 = { 50,100},
6247   "18 = { 50,100},
6248   "1A = { , 50},
6249   "1B = { , 50},
6250   "1C = { 50,100},
6251   "1D = { 50,100},
6252   "1E = { 50,100},
6253   "1F = { 50,100},
6254   "20 = { , 50},
6255   "21 = { , 50},
6256   "22 = { 50,100},
6257   "24 = { , 50},
6258   "27 = { 50,100},

```

```

6259     1 = {100,100},
6260     7 = { 50,100},
6261    "3A = {300,500},
6262    "3B = {200,400},
6263    "3C = {200,100},
6264    "3D = {200,200},
6265    "3E = {100,200},
6266     A = {  ,100},
6267     D = {  , 50},
6268     J = { 50,  },
6269     K = {  , 50},
6270     L = {  , 50},
6271     Q = {  , 50},
6272     T = { 50,  },
6273     X = { 50, 50},
6274     Y = { 50,  },
6275     h = {  , 50},
6276     k = {  , 50}
6277   }
6278

```

Extended by the `eulervm` package.

```

6279 \SetProtrusion
6280   [ name      = euler-vm,
6281     load      = euler ]
6282   { encoding = U,
6283     family   = zeur  }
6284   {
6285     "28 = {100,200},
6286     "29 = {100,200},
6287     "2A = {100,150},
6288     "2B = {100,150},
6289     "2C = {200,300},
6290     "2D = {200,300},
6291     "2E = {  ,100},
6292     "2F = {100,  },
6293     "3F = {150,150},
6294     "5B = {  ,100},
6295     "5E = {100,100},
6296     "5F = {100,100},
6297     "80 = {  , 50},
6298     "81 = {200,250},
6299     "82 = {100,200}
6300   }
6301
6302 </eur>

```

Euler Script font (`euca1`).

```

6303 <*eus>
6304 \SetProtrusion
6305   [ name      = euscript ]
6306   { encoding = U,
6307     family   = eus  }
6308   {
6309     A = {100,100},
6310     B = { 50,100},
6311     C = { 50, 50},
6312     D = { 50,100},
6313     E = { 50,100},
6314     F = { 50,  },
6315     G = { 50,  },
6316     H = {  ,100},
6317     K = {  , 50},

```

```

6318     L = {    ,150},
6319     M = {    , 50},
6320     N = {    , 50},
6321     O = { 50, 50},
6322     P = { 50, 50},
6323     T = {    ,100},
6324     U = {    , 50},
6325     V = { 50, 50},
6326     W = { 50, 50},
6327     X = { 50, 50},
6328     Y = { 50,   },
6329     Z = { 50,100},
6330     "00 = {250,250},
6331     "18 = {200,200},
6332     "3A = {200,150},
6333     "40 = {    ,100},
6334     "5E = {100,100},
6335     "5F = {100,100},
6336     "66 = { 50,   },
6337     "67 = {    , 50},
6338     "6E = {200,200}
6339 }
6340
6341 \SetProtrusion
6342 [ name    = euscript-vm,
6343   load    = euscript ]
6344 { encoding = U,
6345   family   = zeus   }
6346 {
6347   "01 = {600,600},
6348   "02 = {200,200},
6349   "03 = {200,200},
6350   "04 = {200,200},
6351   "05 = {150,150},
6352   "06 = {200,200},
6353   "07 = {200,200},
6354   "08 = {100,100},
6355   "09 = {100,100},
6356   "0A = {100,100},
6357   "0B = {100,100},
6358   "0C = {100,100},
6359   "0D = {100,100},
6360   "0E = {150,150},
6361   "0F = {100,100},
6362   "10 = {150,150},
6363   "11 = {100,100},
6364   "12 = {150,100},
6365   "13 = {100,150},
6366   "14 = {150,100},
6367   "15 = {100,150},
6368   "16 = {200,100},
6369   "17 = {100,200},
6370   "19 = {150,150},
6371   "1A = {150,100},
6372   "1B = {100,150},
6373   "1C = {100,100},
6374   "1D = {100,100},
6375   "1E = {250,100},
6376   "1F = {100,250},
6377   "20 = {150,200},
6378   "21 = {150,200},
6379   "22 = {150,150},
6380   "23 = {150,150},

```



```

6381 "24 = {100,200},
6382 "25 = {150,150},
6383 "26 = {150,150},
6384 "27 = {100,100},
6385 "28 = {100,100},
6386 "29 = {100,150},
6387 "2A = {100,100},
6388 "2B = {100,100},
6389 "2C = {100,100},
6390 "2D = {150,150},
6391 "2E = {150,150},
6392 "2F = {100,100},
6393 "30 = {100,100},
6394 "31 = {100,100},
6395 "32 = {100,100},
6396 "33 = {100,100},
6397 "34 = {100,100},
6398 "35 = {100,100},
6399 "3E = {150,150},
6400 "3F = {150,150},
6401 "60 = { ,200},
6402 "61 = {200, },
6403 "62 = {100,100},
6404 "63 = {100,100},
6405 "64 = {100,100},
6406 "65 = {100,100},
6407 "68 = {300, },
6408 "69 = { ,300},
6409 "6C = {100,100},
6410 "6D = {100,100},
6411 "6F = {100,100},
6412 "72 = {100,100},
6413 "73 = {200,100},
6414 "76 = { ,100},
6415 "77 = {100, },
6416 "78 = { 50, 50},
6417 "79 = {100,100},
6418 "7A = {100,100},
6419 "7D = {150,150},
6420 "7E = {100,100},
6421 "A8 = {100,100},
6422 "A9 = {100,100},
6423 "AB = {200,200},
6424 "BA = { ,200},
6425 "BB = { ,200},
6426 "BD = {200,200},
6427 "DE = {200,200}
6428 }
6429
6430 </eus>

```

Euler Fraktur font (eufrak).

```

6431 <*euf>
6432 \SetProtrusion
6433 [ name = mathfrak ]
6434 { encoding = U,
6435   family = euf }
6436 {
6437   A = { , 50},
6438   B = { , 50},
6439   C = { 50, 50},
6440   D = { , 80},
6441   E = { 50, },

```

```

6442      G = {      , 50},
6443      L = {      , 80},
6444      O = {      , 50},
6445      T = {      , 80},
6446      X = { 80, 50},
6447      Z = { 80, 50},
6448      b = {      , 50},
6449      c = {      , 50},
6450      k = {      , 50},
6451      p = {      , 50},
6452      q = { 50,    },
6453      v = {      , 50},
6454      w = {      , 50},
6455      x = {      , 50},
6456      1 = {100,100},
6457      2 = { 80, 80},
6458      3 = { 80, 50},
6459      4 = { 80, 50},
6460      7 = { 50, 50},
6461      "12 = {500,500},
6462      "13 = {500,500},
6463      ! = {      ,200},
6464      ' = {200,300},
6465      ( = {200,    },
6466      ) = {      ,200},
6467      * = {200,200},
6468      + = {200,250},
6469      - = {200,200},
6470      {,} = {300,300},
6471      . = {400,400},
6472      {=} = {200,200},
6473      : = {      ,200},
6474      ; = {      ,200},
6475      ] = {      ,200}
6476  }
6477
6478  </euf>
6479  </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>).

```

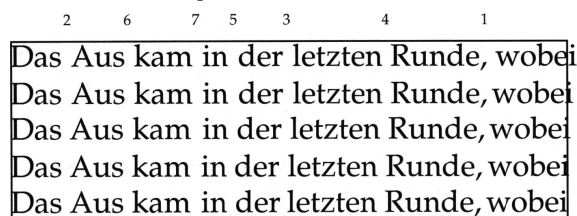
6480  <*cfg-e>
6481  \SetProtrusion
6482  <zpeu|euroitc> { encoding = U,
6483  <mvs> { encoding = {OT1,U},
6484  <zpeu> family = zpeu }
6485  <euroitc> family = {euroitc,euroitcs} }
6486  <mvs> family = mvs }
6487  {
6488  <zpeu> E = {50, }
6489  <euroitc> E = {100,50}
6490  <mvs> 164 = {50,50}, % \EUR
6491  <mvs> 068 = {50,-100} % \EURdig
6492  }
6493
6494  <*zpeu|euroitc>
6495  \SetProtrusion

```

---

<sup>15</sup> 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.



```

6496 { encoding = U,
6497 <zpeu> family = zpeu,
6498 <euroitc> family = {euroitc,euroitcs},
6499 shape = it* }
6500 {
6501 <zpeu> E = {100,-50}
6502 <euroitc> E = {100,}
6503 }
6504
6505 </zpeu|euroitc>
6506 <*zpeu>
6507 \SetProtrusion
6508 { encoding = U,
6509 family = {zpeus,eurosans} }
6510 {
6511 E = {100,50}
6512 }
6513
6514 \SetProtrusion
6515 { encoding = U,
6516 family = {zpeus,eurosans},
6517 shape = it* }
6518 {
6519 E = {200, }
6520 }
6521
6522 </zpeu>
6523 </cfg-e>

```

## 15.9 Interword Spacing

Default unit is space.

```

6524 <*m-t>
6525 %%% -----
6526 %%% INTERWORD SPACING
6527
6528 \SetExtraSpacing
6529 [ name = default ]
6530 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6531 {

```

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

6532           { , } = { , -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)

6533           r = { , -300, 300 },

- [before or] after lowercase characters with ascenders

6534           b = { , -200, 200 },

6535           d = { , -200, 200 },

6536           f = { , -200, 200 },

6537           h = { , -200, 200 },

6538           k = { , -200, 200 },

6539           l = { , -200, 200 },

6540           t = { , -200, 200 },

- [before or] after lowercase characters with x-height plus descender with additional optical space, e. g., ‘v’, or ‘w’

6541           c = { , -100, 100 },

6542           p = { , -100, 100 },

6543           v = { , -100, 100 },

6544           w = { , -100, 100 },

6545           z = { , -100, 100 },

6546           x = { , -100, 100 },

6547           y = { , -100, 100 },

- [before or] after lowercase characters with x-height plus descender without additional optical space

6548           i = { , 50, -50 },

6549           m = { , 50, -50 },

6550           n = { , 50, -50 },

6551           u = { , 50, -50 },

- after colon and semicolon

6552           : = { , 200, -200 },

6553           ; = { , 200, -200 },

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

6554           . = { , 250, -250 },

6555           ! = { , 250, -250 },

6556           ? = { , 250, -250 }

The order has to be reversed when enlarging is needed.’

6557        }

6558

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.?

### 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  $\TeX$ book:

‘If the space factor  $f$  is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if  $f \geq 2000$ . [...] Then the stretch component is multiplied by  $f / 1000$ , while the shrink component is multiplied by  $1000 / f$ .’

The ‘extra space’ (`\fontdimen 7`) for Computer Modern Roman is a third of `\fontdimen 2`, i. e., 333.

```
6559 \SetExtraSpacing
6560   [ name      = nonfrench-cmr,
6561     load       = default,
6562     context    = nonfrench ]
6563   { encoding = {OT1,T1,LY1,OT4,QX,T5},
6564     family    = cmr }
6565   {
```

`latex.ltx` has:

```
\def\nonfrenchspacing{
\sffcode`\. 3000
```

```
6566   . = {333,2000,-667},
```

```
\sffcode`\? 3000
```

```
6567   ? = {333,2000,-667},
```

```
\sffcode`\! 3000
```

```
6568   ! = {333,2000,-667},
```

```
\sffcode`\: 2000
```

```
6569   : = {333,1000,-500},
```

```
\sffcode`\; 1500
```

```
6570   ; = {    , 500,-333},
```

```
\sffcode`\, 1250
```

```
6571   {,}= {    , 250,-200}
```

```
}
```

```
6572   }
```

```
6573
```

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.

```

6574 \SetExtraSpacing
6575 [ name      = nonfrench-default,
6576   load      = default,
6577   context   = nonfrench ]
6578 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6579 {
6580   . = {240,2000,-667},
6581   ? = {240,2000,-667},
6582   ! = {240,2000,-667},
6583   : = {240,1000,-500},
6584   ; = {      , 500,-333},
6585   {,}= {      , 250,-200}
6586 }
6587

```

## 15.10 Additional Kerning

Default unit is 1em.

```

6588 %%% -----
6589 %%% ADDITIONAL KERNING
6590

```

A dummy list to be loaded when no context is active.

```

6591 \SetExtraKerning
6592 [ name = empty ]
6593 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
6594 { }
6595

```

### 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 (T<sub>E</sub>X'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.)

```

6596 \SetExtraKerning
6597 [ name      = french-default,
6598   context   = french,
6599   unit      = space ]
6600 { encoding = {OT1,T1,LY1} }
6601 {
6602   : = {1000,}, % = \fontdimen2
6603   ; = {500, }, % ~ \thinspace
6604   ! = {500, },
6605   ? = {500, }
6606 }
6607

```

<sup>16</sup> [http://fr.wikipedia.org/wiki/Espace\\_typographique](http://fr.wikipedia.org/wiki/Espace_typographique), 5 July 2007.

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>.

```

6608 \SetExtraKerning
6609   [ name      = french-guillemets,
6610     context    = french-guillemets,
6611     load       = french-default,
6612     unit       = space   ]
6613   { encoding = {T1,LY1} }
6614   {
6615     \guillemotleft = { ,800}, % = 0.8\fontdimen2
6616     \guillemotright = {800, }
6617   }
6618
6619 \SetExtraKerning
6620   [ name      = french-guillemets-OT1,
6621     context    = french-guillemets,
6622     load       = french-default,
6623     unit       = space   ]
6624   { encoding = OT1      }
6625   { }
6626
```

### 15.10.2 Turkish

```

6627 \SetExtraKerning
6628   [ name      = turkish,
6629     context    = turkish ]
6630   { encoding = {OT1,T1,LY1} }
6631   {
6632     : = {167, }, % = \thinspace
6633     ! = {167, },
6634     {=} = {167, }
6635   }
6636
6637 </m-t>
6638 </config>

```

## 16 Auxiliary File for Micro Fine Tuning

This file can be used to test protrusion and expansion settings.

```

6639 < *test>
6640 \documentclass{article}
6641
6642 %% Here you can specify the font you want to test, using
6643 %% the commands \fontfamily, \fontseries and \fontshape.
6644 %% Make sure to end all lines with a comment character!
6645 \newcommand*{\TestFont}{%
6646   \fontfamily{ppl}%
6647   \fontseries{b}%
6648   \fontshape{it}% sc, sl
6649 }
6650
6651 \usepackage{ifthen}
6652 \usepackage[T1]{fontenc}
6653 \usepackage[latin1]{inputenc}
6654 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
6655
6656 \pagestyle{empty}
6657 \setlength{\parindent}{0pt}
6658 \newcommand*{\crulefill}{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}

```

```

6659 \newcommand*\testprotrusion[2] [] {%
6660   \ifthenelse{\equal{#1}{r}}{}{#2}%
6661   lorem ipsum dolor sit amet,
6662   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
6663   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6664   you know the rest%
6665   \ifthenelse{\equal{#1}{l}}{}{#2}%
6666   \linebreak
6667   {\fontencoding{\encodingdefault}%
6668   \fontseries{\seriesdefault}%
6669   \fontshape{\shapedefault}%
6670   \selectfont
6671   Here is the beginning of a line, \dotfill and here is its end}\linebreak
6672 }
6673 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
6674 \def\stripprefix#1>{}
6675 \newcount\charcount
6676 \begin{document}
6677
6678 \microtypesetup{expansion=false}
6679
6680 {\centering The font in this document is called by:\
6681 \texttt{\showTestFont}\par}\bigskip
6682
6683 \TestFont\selectfont
6684 This line intentionally left empty\linebreak
6685 %% A -- Z
6686 \charcount=65
6687 \loop
6688   \testprotrusion{\char\charcount}
6689   \advance\charcount 1
6690   \ifnum\charcount < 91 \repeat
6691 %% a -- z
6692 \charcount=97
6693 \loop
6694   \testprotrusion{\char\charcount}
6695   \advance\charcount 1
6696   \ifnum\charcount < 123 \repeat
6697 %% 0 -- 9
6698 \charcount=48
6699 \loop
6700   \testprotrusion{\char\charcount}
6701   \advance\charcount 1
6702   \ifnum\charcount < 58 \repeat
6703 %%
6704 \testprotrusion[r]{,}
6705 \testprotrusion[r]{.}
6706 \testprotrusion[r]{;}
6707 \testprotrusion[r]{:}
6708 \testprotrusion[r]{?}
6709 \testprotrusion[r]{!}
6710 \testprotrusion[l]{\textexclamdown}
6711 \testprotrusion[l]{\textquestiondown}
6712 \testprotrusion[r]{}
6713 \testprotrusion[l]{}
6714 \testprotrusion{/}
6715 \testprotrusion{\char`\}
6716 \testprotrusion{-}
6717 \testprotrusion{\textendash}
6718 \testprotrusion{\textemdash}
6719 \testprotrusion{\textquoteleft}
6720 \testprotrusion{\textquoteright}
6721 \testprotrusion{\textquotedblleft}

```



```
6722 \testprotrusion{\textquotedblright}
6723 \testprotrusion{\quotesinglbase}
6724 \testprotrusion{\quotedblbase}
6725 \testprotrusion{\guilsinglleft}
6726 \testprotrusion{\guilsinglright}
6727 \testprotrusion{\guillemotleft}
6728 \testprotrusion{\guillemotright}
6729
6730 \newpage
6731 The following displays the current font stretched by 5%,
6732 normal, and shrunk by 5\%:
6733
6734 \bigskip
6735 \newlength{\MTln}
6736 \newcommand*{\teststring
6737   {ABCDEFGH IJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxy0123456789}
6738 \settowidth{\MTln}{\teststring}
6739 \microtypesetup{expansion=true}
6740
6741 \parbox{1.05\MTln}{\teststring\linebreak\
6742   \teststring}\par\bigskip
6743 \parbox{0.95\MTln}{\teststring}
6744
6745 \end{document}
6746 </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> ) . . . . .	77	<code>\MT@get@basefamily</code> : only remove suffix if it is ‘x’ or ‘j’ . . . . .	78
issue an error instead of a warning, when pdfTeX version is too old for autoexpand . . . . .	119	<code>\MT@get@listname@</code> : don’t check for empty attributes list . . . . .	79
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i> ) . . . . .	127	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i> ) . . . . .	41
Protrusion: add factors for some more characters settings for Adobe Minion (contributed by <i>Harald Harders</i> ) . . . . .	133	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i> ) . . . . .	83
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance . . . . .	134	<code>\MT@pdfTeX@no</code> : fix concerning version check (reported by <i>Harald Harders</i> ) . . . . .	36
<code>\MT@declare@sets</code> : remove spaces around set name . . . . .	92	<code>\MT@permute</code> : don’t use sets for empty encoding . . . . .	106
<code>\MT@DeclareSet</code> : remove spaces around first argument . . . . .	91	<code>\MT@pr@split</code> : fix: allow zero and negative values . . . . .	57
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded . . . . .	77	<code>\MT@use@set</code> : remove spaces around set name . . . . .	96
		<code>\UseMicrotypeSet</code> : remove spaces around first argument . . . . .	95

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

Font Sets: declare cmor as an alias of cmr . . . . .	126	changed . . . . .	92
new: allmath and basicmath . . . . .	124	<code>\MT@get@inh@list</code> : fix: set inheritance list <code>\globally</code> to <code>\empty</code> . . . . .	81
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding . . . . .	154	<code>\MT@get@listname@</code> : alternatively check for alias font name . . . . .	79
add settings for Computer Modern Roman math symbols . . . . .	159	<code>\MT@get@size</code> : additional magic to catch some errors hijack <code>\set@fontsize</code> instead of <code>\@setfontsize</code> . . . . .	94
<code>\MT@encoding@check</code> : check whether only one encoding specified . . . . .	105	<code>\MT@loop</code> : fix: new macro, used instead of <code>\loop</code> . . . . .	45
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement . . . . .	53	<code>\MT@maybe@do</code> : also check for alias font name . . . . .	53
<code>\MT@get@basefamily</code> : also remove ‘w’ (swash capitals) . . . . .	78	<code>\MT@permute@@@@@</code> : more sanity checks for <code>\SetProtrusion</code> and <code>\SetExpansion</code> . . . . .	107
<code>\MT@get@highlevel</code> : check whether defaults have		<code>\MT@setupfont</code> : also search for alias font file . . . . .	51
		fix: call <code>\@enc@update</code> if necessary . . . . .	51

### Version 1.3 (2004/10/27)

General: fix: specifying load option does no longer require to give a name, too . . . . .	102	<code>\MT@catcodes</code> : check some category codes (compatibility with german) . . . . .	37
Font Sets: declare aer, zer and hfor as an alias of cmr . . . . .	126	<code>\MT@load@list</code> : check whether list exists . . . . .	77

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

General: check for pdfcprot . . . . .	49	(OT1, T1, lmr) . . . . .	138
don’t use scratch registers in global definitions . . . . .	81	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options . . . . .	115
use <code>\pickup@font</code> instead of <code>\define@newfont</code> as the hook for <code>\MT@setupfont</code> . . . . .	86	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too . . . . .	99
use one instead of five counters . . . . .	46		
Protrusion: tweak quote characters for cmr variants			

**Version 1.4a (2004/11/17)**

General: new option: final .....	112	when reading files (reported by <i>Michael Hoppe</i> )	78
\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> ) ..	114	name if encoding failed .....	54
new message if \pdfoutput is changed .....	118	\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	40	\MT@get@slot: don't define \MT@char globally (save stack problem) .....	81
Protrusion: harmonise dashes in upshape and italic (cmr, pad, ppl) .....	133	\MT@ifdimen: don't set \MT@count globally (save stack problem) .....	42
slanted like italics .....	141	\MT@use@set: don't use undeclared font sets .....	96
\MT@checklist@family: fix: don't try alias family			

**Version 1.5 (2004/12/15)**

General: defaults: step: 4 (suggested by <i>Hàn Thê Thành</i> ) .....	112	\MT@catcodes: reset catcode of ‘~’ (compatibility with chemsym) .....	37
defaults: calculate step as min(stretch,shrink)/5	119	\MT@cfg@catcodes: reset catcode of ‘=’ (compatibility with Turkish babel) .....	78
defaults: turn off expansion for DVI output ....	118	\MT@get@highlevel: don't test defaults if called after begin document .....	92
disable automatic expansion for DVI output ....	119	\MT@scale@factor: warning for factors outside limits	59
new option: selected, by default false (suggested by <i>Hàn Thê Thành</i> ) .....	111	\MT@scale@to@em: don't use \lpcode and \rptide for the calculation .....	58
Documentation: add ‘Short History’ .....	29	\MT@set@ex@codes: allow non-selected font expansion .....	63
add note about DVIoutput option .....	8	\MT@set@pr@codes: adjust protrusion factors before setting the inheriting characters .....	56
Inheritance: remove \ss from T1 list, add \DJ ....	128		
Protrusion: settings for Bitstream Charter .....	133		
\DeclareMicrotypeAlias: remove spaces around arguments .....	97		

**Version 1.6 (2005/01/24)**

General: defaults: turn off expansion for old pdfTeX versions .....	113	improve settings for numbers (pointed out by <i>Peter Muthesius</i> ) .....	135
disable automatic expansion for old pdfTeX versions .....	119	tune CMR math letters (OML encoding) .....	159
load a font, if none is active .....	51	\MT@get@charwd: use e-TeX's \fontcharwd, if available	58
new option: factor, by default 1000 .....	112	\MT@get@inh@list: correct message if selected is false .....	81
restructure dtx file .....	124	\MT@set@ex@codes: introduce factor option .....	63
test whether \pickup@font has changed .....	88	\MT@set@pr@codes: introduce factor option .....	56
test whether numeric options receive a number	113	\MT@use@set: retain current set if new set is undeclared .....	96
use e-TeX's \ifcsname and \ifdefined if defined .	40	\MT@vinfo: new macro: used instead of \ifMT@verbose .....	34
Protrusion: add italic uppercase Greek letters ....	141		

**Version 1.6a (2005/02/02)**

Documentation: add table of fonts with tailored protrusion settings .....	21	reported by <i>Bernard Gaulle</i> ) .....	81
\MT@get@slot: completely redone, hopefully more robust (compatible with frenchpro; problem		\MT@pdftex@no: new macro .....	35
		\MT@reset@ef@codes: only reset \efcodes for older pdfTeX versions .....	63

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

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i> ) .....	93	don't allow automatic expansion for old pdfTeX versions .....	104
--	----	---	-----

fix: remove space after autoexpand	104	test for \chardefed commands	82
modify \showhyphens	120	test whether \<encoding>\<...> is defined	82
new value for verbose option: errors	112	\MT@if@list@exists: don't define \MT@#1c@name	
shorter command names	46	\globally, here and elsewhere	80
warning when running in draft mode	118	\MT@if@dimen: comparison with 1 to allow size smaller	
Documentation: add hint about compatibility	26	than 1 (suggested by <i>Andreas Böhmann</i> )	42
remove table of match order	12	\MT@increment: use e-TeX's \numexpr if available	46
Protrusion: fix: remove \ from OT1, add		\MT@is@composite: new macro: construct command	
\textbackslash to T1 encoding	136	for composite character; no uncontrolled expansion	85
\DeclareMicrotypeAlias: may also be used inside		\MT@scale: new macro: use e-TeX's \numexpr if avail-	
configuration files	97	able	46
\LoadMicrotypeFile: new command (suggested by		\MT@set@ex@codes: two versions of this macro	63
<i>Andreas Böhmann</i> )	97	\MT@split@name: don't define \MT@encoding &c.	
\Microtype@Hook: new command for font package		\globally	53
authors	114	\MT@test@ast: make it simpler	92
\microtypesetup: fix: warning also when setting to		\MT@try@order: always check for size, too (suggested	
(no)compatibility	115	by <i>Andreas Böhmann</i> )	79
\MT@begin@catcodes: also use inside configuration		fix: also check for //<series>/<shape>/ (reported	
commands	78	by <i>Andreas Böhmann</i> )	79
\MT@cfg@catcodes: reset catcode of ' (compatibility		\MT@warn@code@too@large: new macro: type out max-	
with french* packages)	78	imum protrusion factor	59
\MT@get@listname@: use \etfor ( <i>Andreas Böhmann's</i>		\MT@warn@err: new macro: for verbose=errors	34
idea)	79		
\MT@get@slot: remove backslash hack	81		

### Version 1.8 (2005/06/23)

General: \SetProtrusion: new key: unit	103	\MT@find@file: no longer wrap names in commands	77
if font substitution has occurred, set up the substi-		\MT@get@charwd: warning for missing (resp. zero-	
tute font, not the selected one	87	width) characters	58
new option: config to load a different main config-		\MT@get@font@dimen@six: new macro: test whether	
uration file	113	\fontdimen 6 is defined	56
new option: unit, by default character	113	\MT@get@listname@: made recursive	79
Documentation: add example for factor option	13	\MT@get@slot: fix: expand active characters	82
add example of how to get rid of a widow (sugges-		test whether \<encoding>\<...> is defined made more	
ted by <i>Adam Kucharczyk</i> )	15	robust	82
add hint about error messages	26	\MT@get@unit: new macro: get unit for codes	60
Font Sets: add U encoding to allmath	124	\MT@in@rlist: made recursive	45
declare pxr and txr as aliases of ppl resp. ptm	126	\MT@is@active: new macro: translate inputenc-	
Inheritance: remove \DJ from T1 list (it's the same as		defined characters	83
\DH)	128	\MT@is@letter: warning for non-ASCII characters	82
Protrusion: add LY1 characters for Times	140	\MT@led@kern: character protrusion with ledmac	48
settings for AMS math fonts	162	\MT@map@clist@n: new macro: used instead of \etfor	44
verified settings for slanted Computer Modern Ro-		\MT@map@tlist@n: new macro: used instead of \etfor	44
man	147	\MT@old@cmd: renamed commands from	
\add@accent: fix: disable micro-typographic setup in-		\..MicroType.. to \..Microtype..	33
side \add@accent (reported by <i>Stephan Hennig</i> )	88	\MT@pdf@tex@no: case 5: pdfTeX 1.30	36
\DeclareMicrotypeAlias: warning when overriding		\MT@permute@@@@@: add ranges to the beginning of	
an alias font	97	the lists	107
\DeclareMicrotypeSetDefault: new command: set		\MT@pr@split: get character width once only	57
default font set	96	\MT@scale: fix: remove spaces in e-TeX variant (repor-	
\MT@cfg@catcodes: reset catcodes of the remaining		ted by <i>Mark Rossi</i> )	46
ASCII characters	78	\MT@setupfont@hook: restore \% and \# when	
\MT@check@rlist: made recursive	108	hyperref is loaded	49
\MT@curr@list@name: new macro: current list type		restore csquotes's active characters	49
and name	85	restore percent character if Spanish babel is loaded	49
\MT@declare@sets: warning when redefining a set	92	\MT@use@set: fix: remove braces in first line	96
\MT@define@set@key@: use comma lists instead of		\MT@xadd: simplified	43
token lists	92		

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

General: <code>\DeclareMicrotypeSet</code> : new key: font ..	94	settings for OT4 encoding (Computer Modern Roman, Palatino, Times) .....	133
<code>\SetProtrusion</code> : value ‘relative’ renamed to ‘character’ for key unit .....	103	settings for T5 encoded Computer Modern Roman .....	133
allow context-specific font setup .....	87	<code>\DisableLigatures</code> : new command: disable ligatures (requires pdf $\TeX$ 1.30) .....	98
disable expansion if both step and shrink are zero .....	120	<code>\microtypecontext</code> : new command: change setup context in the document .....	89
disable microtype setup inside hyperref’s <code>\pdfstringdef</code> (reported by <i>Hàn Thê Thành</i> ) ..	50	<code>\MT@checklist@family</code> : fix: add two missing <code>\expandafters</code> .....	54
fix: use true as the default value .....	110	<code>\MT@detokenize@c</code> : fix the $\mathbb{X}$ version .....	41
option unit: rename value relative to character .....	113	<code>\MT@exp@two@n</code> : new macros: less <code>\expandafters</code> ..	40
warning if user requested zero step .....	119	<code>\MT@get@opt</code> : new key ‘preset’ to set all characters to the specified value before loading the lists ..	60
Documentation: add hint about verbatim environment .....	26	<code>\MT@is@active</code> : redone: use <code>\set@display@protect</code> ..	83
add remark about Type 1 fonts required for automatic font expansion .....	7	<code>\MT@is@letter</code> : using <code>\catcode</code> should be more efficient than inspecting the <code>\meaning</code> .....	82
Font Sets: add OT4 encoding to text sets .....	124	<code>\MT@maybe@do</code> : redone .....	53
add T5 encoding to text sets .....	124	<code>\MT@pdf@tex@no</code> : compatibility with $\TeX$ Live hack (reported by <i>Herbert Voß</i> ) .....	36
declare <code>qpl</code> and <code>qtm</code> (qfonts, $\TeX$ Gyre) as aliases of <code>ppl</code> resp. <code>ptm</code> .....	126	<code>\MT@rem@from@clist</code> : new macro: remove an item from a comma list .....	44
Inheritance: add list for OT4 .....	129	<code>\MT@scale@factor</code> : generalised .....	59
add list for T5 (requested by <i>Hàn Thê Thành</i> ) ..	130	<code>\MT@toks</code> : use instead of <code>\toks@</code> .....	38
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR .....	137	<code>\SetProtrusion</code> : (et al.) new key: font .....	98

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

General: ‘ <code>(file name)/(line number)</code> ’ as default list name .....	102	diately (requested by <i>Georg Verwey</i> ) .....	92
new option: <code>defersetup</code> , by default true .....	111	<code>\MT@get@highlevel</code> : no longer check whether defaults have changed .....	92
remove superfluous test whether <code>\pickup@font</code> has changed .....	88	<code>\MT@ifdefined@c@T</code> : new macros: true case only ..	41
Documentation: add explanation for error message in DVI mode .....	26	<code>\MT@ifint</code> : use <code>\pdfmatch</code> if available .....	41
add explanation for error message with non-Type 1 fonts .....	27	<code>\MT@ifstreq</code> : use <code>\pdfstrcmp</code> if available .....	43
Font Sets: declare <code>mbch</code> (mathdesign) as an alias of Charter .....	126	<code>\MT@in@clist</code> : fix .....	44
Protrusion: fix: remove <code>_</code> from OT1 encoding ..	137	<code>\MT@info@missing@char</code> : info instead of warning (after <i>Michael Hoppe</i> reported that the ‘fl’ ligature is missing in Palatino SC) .....	59
settings for T5 encoded Charter .....	133	<code>\MT@is@feature</code> : new macro: check for pdf $\TeX$ feature .....	46
<code>\microtypesetup</code> : inside the preamble, accepts all package options .....	115	<code>\MT@map@clist@n</code> : following $\LaTeX$ 3 .....	44
<code>\MT@check@font@cx</code> : optimise context-sensitive setup ..	89	<code>\MT@permute@@@@</code> : don’t define permutations for unused encodings .....	107
<code>\MT@define@set@key@</code> : don’t expand variables immediately (requested by <i>Georg Verwey</i> ) .....	92	<code>\MT@rem@from@clist</code> : fix .....	44
		<code>\MT@setup@</code> : defer setup until the end of the preamble ..	47

**Version 1.9b (2006/01/20)**

General: compatibility with <code>listings</code> : sanitise more catcodes (reported by <i>Holger Uhr</i> ) .....	50	add samples of micro-typographic features .....	4
compatibility with the <code>extendedchar</code> option of the <code>listings</code> package .....	50	<code>\MT@features</code> : use throughout the package to adjust to beta-ness .....	46
Documentation: activate expansion in the distributed PDF .....	1	<code>\MT@ifdimen</code> : use <code>\pdfmatch</code> if available .....	42
		<code>\MT@warn@code@too@large</code> : fix calculation with present factor .....	59

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

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verwey</i> )	22	<code>\MT@define@code@key@font</code> : fix: context was ignored	101
Protrusion: settings for URW Garamond	134	<code>\MT@define@code@key@size</code> : fix: embrace <code>\MT@tempsize</code> in <code>\csname</code> (bug introduced in 1.9b)	101

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

Font Sets: <code>md*</code> instead of <code>m</code> series in basic sets	124	<code>\MT@get@font@dimen</code> : warning for zero <code>fontdimen</code>	59
add QX encoding to text sets	124	<code>\MT@get@opt</code> : optimise: don't reset when preset option is set	60
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i> )	130	set list name before presetting	60
Protrusion: settings for QX encoding (contributed by <i>Maciej Eder</i> )	139	<code>\MT@is@active</code> : support for Unicode (inputenc/utf8)	83
settings for Euro symbols (Adobe, ITC, marvosym)	170	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>tex4ht</code> is loaded (reported by <i>Peter Dybala</i> )	49
tweak AMS settings	162	<code>\SetProtrusion</code> : (et al.) optimise: unify keys for mandatory argument	98
<code>\DeclareCharacterInheritance</code> : fix: empty context	104	(et al.) split keys of optional and mandatory argument	98
<code>\MT@detokenize@n</code> : new macro: use <code>\detokenize</code> if available	41		
<code>\MT@get@ex@opt</code> : fix: evaluate preset	64		

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

General: fix: default value for <code>activate</code> : true	110	settings for Euler Roman font	166
Documentation: add hint about unknown encodings	25	<code>\DeclareCharacterInheritance</code> : new key 'inputenc' to set the input encoding	104
include LPPL	192	<code>\MT@rem@from@clist</code> : model after <code>\@removeelement</code>	44
Font Sets: declare <code>zeur</code> and <code>zeus</code> ( <code>eulervm</code> ) as aliases of <code>eur</code> resp. <code>eus</code> ( <code>euler</code> )	126	<code>\MT@setup@</code> : empty <code>\MT@setup@</code> after use (compatibility with the <code>combine</code> class)	47
Inheritance: adapt to <code>marvosym</code> 's changed encoding	131	<code>\pickup@font</code> : no tracing with <code>trace</code> package	87
Protrusion: complete settings for Euler Fraktur and Script fonts	169	<code>\SetExpansion</code> : new key: <code>inputenc</code>	99
fix: forgotten comma in <code>mt-mvs.cfg</code> ; adapt to <code>marvosym</code> 's changed encoding	170	<code>\SetProtrusion</code> : (et al.) new key: <code>inputenc</code>	98

**Version 1.9f (2006/09/09)**

Protrusion: fix: <code>euler-vm</code> did not load <code>euler</code> settings	167	<code>\MT@reset@context</code> : only reset context if it has actually been changed	90
<code>\MT@curr@list@name</code> : fix: <code>\MessageBreak</code> must not be expanded	85	<code>\MT@set@inh@list</code> : fix: forgotten comma in the features list	105
<code>\MT@gdef@n</code> : new macros: global variants	40	<code>\MT@set@named@keys</code> : new macro: set name first, simplify parsing of optional argument	100
<code>\MT@get@inh@list</code> : fix: input encoding must be set after the inheritance list has been parsed	81	<code>\SetProtrusion</code> : (et al.) set catcodes before parsing optional argument	98
<code>\MT@glet</code> : new macro	40		

**Version 2.0 (2007/01/14)**

General: (beta:1) new option: <code>babel</code> , by default false (language-dependent setup suggested by <i>Ulrich Dirr</i> )	111	shorthands	124
(beta:1) new option: <code>letterspace</code> , by default 100	112	compatibility with listings: set catcode of backslash to zero (reported by <i>Steven Bath</i> )	50
(beta:3) compatibility with <code>soul</code> : register <code>\textls</code> and <code>\sstyle</code>	50	maybe disable <code>\MT@noligatures</code> after the preamble	123
(beta:8) option 'babel': fix: switch off French <code>babel</code> 's shorthands with <code>\shorthandoff</code> instead of <code>\NoAutoSpaceBeforeFDP</code> (reported by <i>Daniel Flipo</i> )	124	new package <code>letterspace</code> : a stripped-down version, containing the <code>letterspacing</code> commands only	1
(beta:8) option 'babel': switch off Turkish <code>babel</code> 's shorthands		option 'unit', <code>\SetProtrusion</code> : deprecate value 'relative' completely	103
		option 'letterspacing' renamed to 'letterspace'	112
		Documentation: add hint about how to increase	



font_max and font_mem_size .....	27	totally redone, using the new \letterspacefont	71
add hint about warning when tracking <i>and</i> expansion is applied to a font .....	27	\MT@declare@sets: fix: empty size list when redefining set .....	92
add remark about ‘draft’ option disabling microtype (noted by <i>Michalis Miatidis</i> ) .....	9	\MT@is@symbol: made even more robust .....	84
qualify hint about web documents with regard to older pdfTeX versions .....	25	\MT@load@inputenc: sanitise catcodes before loading input encoding (problem with listings) ....	61
qualify hints about expansion error messages with regard to older pdfTeX versions .....	27	\MT@pdfTeX@no: (beta:1) case 6: pdfTeX 1.40 .....	36
Font Sets: add footnotesize and scriptsize sets	124	\MT@split@name: adjust to possible letterspacing ..	53
add smallcaps set .....	124	\SetExtraKerning: (beta:1) new command: additional kerning .....	100
Kerning: remove letterspacing settings .....	175	\SetExtraSpacing: (beta:1) new command: adjustment of interword spacing .....	100
\DeclareMicrotypeBabelHook: (beta:1) new command: interaction with babel .....	98	\SetTracking: new command: tracking .....	99
\lststyle: (beta:1) new command: letterspacing ..	71	\textls: (beta:1) new command: letterspacing ...	75
(beta:8) fix: font switches don’t pose a problem anymore .....	71	starred version: remove spaces around text ....	75
(beta:8) fix: letterspacing commands may be nested .....	71	\tracingmicrotypeinpdf: new debug method: mark all fonts with PDF annotations .....	34

### Version 2.1 (2007/01/21)

General: compatibility with CJK: also check for its definition .....	87	\slig: new command: protect ligatures in letter-spaced text .....	72
compatibility with pinyin: disable microtype in \py@macron (reported by <i>Sven Naumann</i> ) ....	50	\MT@get@ls@basefont: redone: use \pdfmatch to make it bullet-proof .....	72
fix: letterspace package forgot to load keyval ..	38	\textls: fix: use \hmode@bgroup .....	75

### Version 2.2 (2007/07/14)

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i> ) .....	47	\MT@get@ls@basefont: fix again: remember base font in a macro .....	72
new option: copyfonts .....	111	\MT@ifdimen: employ LuaTeX features if available ..	42
simplify key declarations .....	102	\MT@ifint: employ LuaTeX features if available ...	41
warning if stretch or shrink aren’t multiples of step .....	120	\MT@ifstreq: employ LuaTeX features if available ..	43
Documentation: add hint about error message with pdfTeX 1.40 .....	27	fix: <del>ε</del> -TeX version shouldn’t use \x and \y (found by <i>Wiebke Petersen</i> ) .....	43
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i> ) .....	26	\MT@is@composite: more robust: expand exactly once	85
add overview .....	6	\MT@is@symbol: expand once more (for frenchpro)	84
logo transparency and amusement .....	1	\MT@lsfont: use \font@name, not \MT@font .....	69
Font Sets: declare FPL Neu as an alias of Palatino ..	126	\MT@maybe@etex: use catcode trickery .....	39
declare chr (chmath) as an alias of Charter (reported by <i>Geoff Vallis</i> ) .....	126	\MT@pdfTeX@no: case 7: pdfTeX 1.40.4 .....	36
default set for tracking: smallcaps .....	125	\MT@preset@aux@space: generalised .....	62
Inheritance: remove ‘-’ → ‘127’ .....	128	\MT@requires@luatex: (basic) support for LuaTeX ..	39
Protrusion: settings for Bitstream Letter Gothic ..	133	\MT@set@all@pr: (et al.) allow empty values .....	56
Spacing: add sample .....	171	\MT@set@inputenc@: only load inputenc files if necessary .....	61
Tracking: add ligatures that are to be disabled ....	131	\MT@set@tr@codes: disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	70
\DeclareMicrotypeVariants: new command .....	97	possibility to customise interword spacing .....	69
\DisableLigatures: new optional argument: disable selected ligatures only .....	98	\MT@setupfont: don’t call \@@enc@update anymore ..	51
\slig: always defined .....	72	only add features that are available with the respective pdfTeX .....	51
\MT@checklist@font: fix: construct font name from characteristics .....	55	\MT@setupfont@hook: restore percent character if Galician babel is loaded .....	49
\MT@copy@font: optionally work on copies of fonts ..	52	\MT@the@pr@code@tr: adjust protrusion of letter-spaced fonts .....	57
\MT@get@basefamily: redone, working on font names and suffixes of arbitrary length .....	79	\MT@tracking: remember fonts that shouldn’t be letterspaced .....	69
\MT@get@charwd: subtract letterspacing amount from width .....	58	\MT@tracking@: fix: tracking couldn’t be re-enabled	69

<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode .....	121	new keys ‘spacing’ and ‘outer spacing’ to adjust interword spacing (suggested by <i>Steven E. Harris</i> ) .....	99
<code>\MT@with@babel@and@T</code> : also inspect class options ..	47	third argument may be empty .....	99
<code>\pickup@font</code> : letterspace: setup inside group ...	87	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code> .....	90
<code>\SetTracking</code> : new key ‘no ligatures’ to disable ligatures of letterspaced fonts .....	99		

### Version 2.3 (2007/12/23)

General: disable <code>\microtypecontext</code> in hyperref’s <code>\pdfstringdef</code> .....	50	<code>\MT@in@clist</code> : fix: don’t use <code>\x</code> (reported by <i>Peter Meier</i> ) .....	44
enable protrusion when tracking is enabled ....	121	<code>\MT@is@active</code> : support for extended Unicode (inputenc/utf8x resp. ucs) – experimental ...	83
fix: really switch off Turkish shorthands .....	124	<code>\MT@no@ligatures</code> : fix: set evaluation didn’t work (bug introduced in 2.2) .....	76
new value for verbose option: silent (suggested by <i>Karl Berry</i> ) .....	112	<code>\MT@plain</code> : letterspace: support for eplain/miniltx ..	35
turned some warnings into errors .....	112	<code>\MT@set@curr@os</code> : adjusting spaces made more reliable .....	70
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> ) .....	71
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> ) .....	88	<code>\MT@SetTracking</code> : sanity check for value .....	99
<code>\lslig</code> : redone: extract outer kerns from current letterspacing amount .....	72	<code>\MT@tr@outer@l</code> : only change pre outer space if it contains glue .....	74
<code>\microtypecontext</code> : made robust (reported by <i>Stephan Hennig</i> ) .....	89		
<code>\MT@begin@catcodes</code> : fix: don’t disable <code>\KV@sp@def</code> ..	78		
<code>\MT@define@set@key@font</code> : font: single asterisk means normal font .....	94		

## B Index

Links refer to the page where the corresponding entry is described (bold face) resp. occurs. Plain numbers refer to the code line where the corresponding entry is defined (underlined) resp. used.

<b>Options:</b>	<code>\DeclareMicrotypeSet*</code> .....	10
DVIoutput .....	<code>\DeclareMicrotypeSetDefault</code> .....	12
activate .....	<code>\DeclareMicrotypeVariants</code> .....	20
auto .....	<code>\DeclareMicrotypeVariants*</code> .....	20
babel .....	<code>\DisableLigatures</code> .....	24
config .....	<code>\LoadMicrotypeFile</code> .....	21
draft .....	<code>\lslig</code> .....	24
expansion .....	<code>\lsstyle</code> .....	23
factor .....	<code>\microtypecontext</code> .....	22
final .....	<code>\microtypesetup</code> .....	9
kerning .....	<code>\SetExpansion</code> .....	14
letterspace .....	<code>\SetExtraKerning</code> .....	19
protrusion .....	<code>\SetExtraSpacing</code> .....	18
selected .....	<code>\SetProtrusion</code> .....	12
shrink .....	<code>\SetTracking</code> .....	16
spacing .....	<code>\textls</code> .....	23
step .....	<code>\textls*</code> .....	23
stretch .....	<code>\textmicrotypecontext</code> .....	22
tracking .....	<code>\UseMicrotypeSet</code> .....	11
unit .....		
verbose .....	<code>\@enc@update</code> .....	791

### User Commands:

<code>\DeclareCharacterInheritance</code> .....	20	<b>A</b>
<code>\DeclareMicrotypeAlias</code> .....	21	aoposter (package) .....
<code>\DeclareMicrotypeBabelHook</code> .....	23	activate (option) .....
<code>\DeclareMicrotypeSet</code> .....	10	<code>\add@accent</code> .....
		ae (package) .....

94
6, 110, 182
2333
21, 126



- amssymb (package) ..... 162  
 article (package) ..... 22  
 auto (option) ..... 7, 111
- B**
- babel (option) ..... 9, 23, 29, 111, 182  
 babel (package) ..... 3, 5, 17, 23, 24, 48,  
     49, 78, 98, 123, 124, 126, 179, 180, 182, 183
- C**
- \cf@encoding ..... 791  
 chemsym (package) ..... 179  
 chmath (package) ..... 21, 126, 183  
 CJK (package) ..... 50, 87, 183  
 cm-super (package) ..... 7  
 color (package) ..... 9, 118  
 combine (package) ..... 47, 182  
 config (option) ..... 9, 20, 30, 113, 180  
 contour (package) ..... 118  
 copyfonts (option) ..... 52, 111, 183  
 crop (package) ..... 118  
 csquotes (package) ..... 28, 30, 48, 49, 137, 180  
 \curr@fontshape ..... 2289, 2291, 2331
- D**
- \DeclareCharacterInheritance ..... 15, 20, 3042  
 \DeclareMicrotypeAlias ..... 17, 21, 44, 2715  
 \DeclareMicrotypeBabelHook ..... 19, 23, 38, 2768  
 \DeclareMicrotypeSet ..... 6, 10, 31, 33, 45, 2442  
 \DeclareMicrotypeSet\* ..... 10, 2442  
 \DeclareMicrotypeSetDefault ..... 8, 12, 35, 2674  
 \DeclareMicrotypeVariants ... 16, 20, 32, 37, 2702  
 \DeclareMicrotypeVariants\* ..... 20  
 defersetup (option) ..... 49, 111, 115, 181  
 \define@newfont ..... 2288, 2290, 2293  
 \DisableLigatures ..... 14, 24, 36, 2748  
 \do@subst@correction ..... 2330  
 docstrip (package) ..... 32  
 draft (option) ..... 9, 111, 112, 183  
 dsfont (package) ..... 56  
 DVIoutput (option) ..... 8, 9, 30, 111, 179
- E**
- eco (package) ..... 21, 126  
 \efcode ..... 1282, 1316, 1317, 1361, 1363  
 eplain (package) . 1, 24, 29, 32, 35, 47, 72, 117, 184  
 eucal (package) ..... 167  
 eufrak (package) ..... 169  
 euler (package) ..... 85, 166, 182  
 eulervm (package) ..... 21, 126, 167, 182  
 euroitc (package) ..... 170  
 europs (package) ..... 170  
 eurosans (package) ..... 170  
 expansion (option) ..... 6, 7, 11, 110, 118
- F**
- \f@encoding ..... 791  
 \f@size ..... 2289, 2291, 2331  
 factor (option) ..... 7, 13, 31, 113, 179  
 fancyvrb (package) ..... 26  
 final (option) ..... 9, 31, 111, 112, 179  
 fix-cm (package) ..... 53  
 \font ..... 780, 1682
- \font@name ..... 83, 818,  
     820, 822, 824, 836, 837, 1579, 1584, 1608,  
     1614, 1624, 1627, 1670, 1682, 1694, 1696,  
     1701, 1703, 1706, 1735, 1867, 1869, 2288,  
     2290, 2293, 2314, 2316, 2318, 2332, 2344, 2364  
 \fontcharwd ..... 1044  
 \fontdimen ..... 972, 979, 1066, 1071, 1172,  
     1175, 1597, 1598, 1614, 1694, 1729, 1737,  
     1743, 1747, 1754, 1764, 1768, 1771, 1880, 1883  
 fontinst (package) ..... 174  
 fontinstallationguide (package) ..... 152  
 french (package) ..... 180  
 frenchpro (package) ..... 84, 179, 183
- G**
- german (package) ..... 37, 178  
 graphics (package) ..... 9, 118
- H**
- hfoldsty (package) ..... 21, 126  
 hyperref (package) . 9, 49, 50, 87, 118, 180, 181, 184
- I**
- IEEEtran (package) ..... 87  
 \iffontchar ..... 1077  
 \ifMT@auto ..... 214, 3706, 3751  
 \ifMT@babel ..... 214, 3870, 3917  
 \ifMT@do ..... 856, 917, 1559, 1899  
 \ifMT@document ..... 256, 3160  
 \ifMT@draft ..... 214, 3640  
 \ifMT@expansion ..... 214, 3670, 3748  
 \ifMT@if@ ..... 213, 2358, 3264, 3937, 3940  
 \ifMT@inannot ..... 72  
 \ifMT@inlist@ ..... 518, 563, 624,  
     647, 797, 884, 901, 910, 928, 946, 1557,  
     1950, 1961, 2010, 2058, 2352, 2416, 2428, 2733  
 \ifMT@kerning ..... 214, 3812, 3859, 3930  
 \ifMT@noligatures ..... 214, 3886  
 \ifMT@nonselected ..... 1260, 2071, 2079  
 \ifMT@norest ..... 2133, 2140  
 \ifMT@opt@auto ..... 3271, 3709  
 \ifMT@opt@DVI ..... 3271, 3656  
 \ifMT@opt@expansion ..... 3271, 3651  
 \ifMT@protrusion ..... 214, 653, 675, 3658, 3797  
 \ifMT@selected ..... 214, 3734, 3754  
 \ifMT@spacing ..... 214, 3803, 3853, 3869  
 \ifMT@tracking ..... 214, 3787, 3847  
 ifpdf (package) ..... 9, 118  
 \iftracingmicrotypeinpdfall ..... 77  
 inputenc (package) .....  
     .. 14, 26, 30, 50, 61, 82, 83, 85, 180, 182–184
- J**
- jurabib (package) ..... 87
- K**
- kerning (option) ..... 7, 11, 29, 110  
 keyval (package) ..... 38, 57, 127, 183  
 \naccode ..... 1510, 1511, 1523, 1526, 1532  
 \knbccode ..... 1503, 1504, 1522, 1525, 1531  
 \knbscode ..... 1406, 1407, 1432, 1436, 1442

## L

ledmac (package) ..... 28, 30, 48, 87, 180  
 ledpar (package) ..... 28  
 \leftmarginkern ..... 661  
 letterspace (option) 8, 18, 23, 24, 29, 42, 69, 113, 182  
 letterspace (package) ..... 24  
 \letterspacefont ..... 1584  
 lineno (package) ..... 48  
 listings (package) ..... 26, 50, 61, 78, 181–183  
 lmodern (package) ..... 7, 139  
 \LoadMicrotypeFile ..... 18, 21, 47, 2729, 3645  
 \lcode ..... 985, 1017, 1018, 1208, 1211, 1636  
 \lslig ..... 24, 28, 1687, 1690  
 \lststyle 23, 25, 733, 754, 1671, 1829, 1833, 1847, 3648

## M

marvosym (package) ..... 21, 30, 32, 131, 170, 182  
 mathdesign (package) ..... 21, 126, 181  
 memoir (package) ..... 22, 87  
 \MicrotypeHook ..... 114, 3498  
 \microtypecontext .....  
 .. 21, 22, 731, 2386, 2396, 3647, 3900, 3903  
 \microtypesetup ..... 9, 20, 3502, 3646  
 miniltx (package) .... 1, 24, 29, 32, 35, 47, 51, 184  
 minimal (package) ..... 51  
 \MT@efont ..... 312, 800, 816, 947,  
 950, 974, 1067, 1081, 1569, 2013, 2084, 2319  
 \MT@abbr@ex ..... 600  
 \MT@abbr@ex@cx ..... 600  
 \MT@abbr@ex@inh ..... 600  
 \MT@abbr@ex@kn ..... 600  
 \MT@abbr@ex@kn@cx ..... 600  
 \MT@abbr@ex@kn@inh ..... 600  
 \MT@abbr@ex@nl ..... 600  
 \MT@abbr@ex@pr ..... 600  
 \MT@abbr@ex@pr@cx ..... 600  
 \MT@abbr@ex@pr@inh ..... 600  
 \MT@abbr@ex@sp ..... 600  
 \MT@abbr@ex@sp@cx ..... 600  
 \MT@abbr@ex@sp@inh ..... 600  
 \MT@abbr@ex@tr ..... 600  
 \MT@abbr@ex@tr@cx ..... 600  
 \MT@active@features .....  
 826, 2346, 2349, 2361, 2368, 2405, 2415,  
 2755, 3659, 3749, 3794, 3804, 3813, 3892, 3893  
 \MT@addto@annot ..... 72  
 \MT@addto@setup ..... 639, 694,  
 776, 1185, 2282, 2386, 2387, 2952, 3373,  
 3503, 3639, 3785, 3846, 3885, 3891, 3916, 3949  
 \MT@auto ..... 1269, 3705, 3718, 3727, 3731  
 \MT@auto@ ..... 1269, 1278, 1334  
 \MT@autofalse ..... 216, 3716, 3726  
 \MT@autotruer ..... 216, 3462  
 \MT@babel@false ..... 223  
 \MT@babel@true ..... 223  
 \MT@begin@catcodes 1951, 1952, 1992, 2737, 2750,  
 2775, 2789, 2811, 2830, 2844, 3047, 3475, 3476  
 \MT@cat ..... 1181, 2274, 2276  
 \MT@catcodes ..... 187, 207  
 \MT@cfg@catcodes ..... 741, 1203, 1977, 1994

\MT@char ..... 1007, 1017, 1018, 1023,  
 1024, 1027, 1029, 1044, 1045, 1077, 1208–  
 1212, 1304, 1316, 1317, 1319, 1321, 1361–  
 1363, 1395, 1406, 1407, 1413, 1414, 1420,  
 1421, 1424, 1426, 1432–1437, 1492, 1503,  
 1504, 1510, 1511, 1514, 1516, 1522–1526,  
 1908, 1909, 2113, 2129, 2130, 2135, 2206,  
 2209, 2211, 2237, 2240, 3107, 3108, 3113, 3116  
 \MT@char@ .....  
 2113, 2119, 2124, 2129, 2143, 2145, 2151,  
 2152, 2154, 2167, 2168, 2171, 2172, 2175,  
 2176, 2180, 2182, 2210, 2223, 2226, 2229, 2248  
 \MT@charstring ..... 2126, 2209, 2214  
 \MT@check@active@set .....  
 ..... 3490, 3665, 3769, 3796, 3807, 3817  
 \MT@check@font ..... 796, 2343, 2407  
 \MT@check@font@cx ..... 2347, 2407  
 \MT@check@font@range ..... 3235, 3237  
 \MT@check@font@range@ ..... 3237, 3238  
 \MT@check@font@list ..... 3188, 3228  
 \MT@check@font@list@ ..... 3228, 3229  
 \MT@check@font@step ..... 3757  
 \MT@checklist@ ..... 864, 877, 1896  
 \MT@checklist@family ..... 895  
 \MT@checklist@font ..... 939  
 \MT@checklist@size ..... 923  
 \MT@checksetup ..... 3512, 3523, 3530, 3550, 3569  
 \MT@clear@options ..... 161, 170, 298, 635, 3634  
 \MT@clist@break ..... 490, 890, 918, 934, 948  
 \MT@clist@function ..... 490  
 \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 3465, 3473, 3474, 3478, 3479, 3482  
 \MT@context ..... 2034, 2046, 2065, 2092  
 \MT@copy@font 785, 813, 2949, 2954, 3390, 3392, 3755  
 \MT@copy@font@ ..... 813, 2949, 2954, 3390, 3755  
 \MT@count ..... 428, 429, 582, 1037, 1044, 1046,  
 1047, 1052, 1057, 1058, 1062, 1071, 1104, 1166  
 \MT@curr@file ..... 1955, 1956,  
 1965, 1966, 2739, 2740, 2923, 3069, 3478, 3951  
 \MT@curr@list@name ..... 1069, 1107, 1197,  
 1330, 1936, 2103, 2243, 2251, 2256, 2264, 2270  
 \MT@curr@ls ..... 1605, 1691, 1694  
 \MT@curr@set@name .....  
 2467, 2469–2472, 2474, 2475, 2480, 2485,  
 2489, 2490, 2515, 2519, 2548, 2579, 2585,  
 2591, 2595, 2596, 2826, 2884, 2902, 2918,  
 2930, 2977, 2981, 2986, 2989, 2991, 2995,  
 2998, 3001, 3011, 3024, 3030, 3038, 3070, 3072  
 \MT@declare@char@inh ..... 3052, 3057, 3067  
 \MT@declare@sets ..... 2450, 2456, 2468, 2757  
 \MT@DeclareSet ..... 2445, 2447, 2464  
 \MT@DeclareSetAndUseIt ..... 2444, 2463  
 \MT@DeclareVariants ..... 2705, 2706, 2708  
 \MT@def@bool@opt ..... 3344,  
 3357–3359, 3368, 3380, 3388, 3397, 3405  
 \MT@def@n ..... 303, 2438, 2439  
 \MT@default@ex@set ..... 2690  
 \MT@default@kn@set ..... 2690  
 \MT@default@pr@set ..... 2690  
 \MT@default@sp@set ..... 2690

- \MT@default@tr@set ..... 2690  
 \MT@define@code@key . 2866, 2910–2913, 3092–3094  
 \MT@define@code@key@font ..... 2889, 2915, 3096  
 \MT@define@code@key@size ..... 2877, 2914, 3095  
 \MT@define@opt@key .. 2916, 2932–2935, 3034–3036  
 \MT@define@optionX ..... 3502, 3559, 3560  
 \MT@define@optionX@ ..... 3561, 3588, 3590, 3591  
 \MT@define@set@key@ ..... 2478, 2637–2640  
 \MT@define@set@key@font ..... 2583, 2642  
 \MT@define@set@key@size ..... 2508, 2641  
 \MT@detokenize@c ..... 346, 2121, 2207  
 \MT@detokenize@n ..... 346, 2239  
 \MT@dimen@six 971, 1038, 1062, 1104, 1763, 1766, 1876  
 \MT@edinfo ..... 60  
 \MT@edinfo@nl ..... 60  
 \MT@DisableLigatures ..... 2748  
 \MT@do@font ..... 580, 987, 1282, 1445, 1533, 1636  
 \MT@documentfalse ..... 256  
 \MT@documenttrue ..... 256, 3895  
 \MT@dofalse ..... 856, 868, 889, 906, 933, 951  
 \MT@dotrue ..... 856, 859, 886, 903, 912, 930, 1891  
 \MT@draftfalse ..... 219, 3407  
 \MT@drafttrue ..... 219, 3409  
 \MT@edef@n ..... 305, 2070, 2096,  
 2432, 2872, 2923, 2925, 3068, 3083, 3215, 3437  
 \MT@encoding ..... 846, 943,  
 2029, 2041, 2121, 2207, 2238, 2255, 2262, 2270  
 \MT@encoding@check ..... 3081, 3085  
 \MT@end@catcodes ..... 1953, 1996, 2745,  
 2759, 2786, 2808, 2827, 2841, 2855, 3062, 3477  
 \MT@error .. 49, 290, 627, 697, 1928, 1941, 2546,  
 2669, 2696, 2942, 2957, 2967, 3382, 3399, 3912  
 \MT@ex@cc@name ..... 1253,  
 1255, 1336, 1337, 1347, 1353, 1354, 1366, 2792  
 \MT@ex@context ..... 816, 2399, 2437  
 \MT@ex@do ..... 1256, 1294  
 \MT@ex@doc@contexts ..... 2437  
 \MT@ex@factor ..... 224, 1270, 1340  
 \MT@ex@factor@ . 1270, 1284, 1287, 1306, 1307, 1334  
 \MT@ex@inh@name ..... 1318, 1319, 1321  
 \MT@ex@level ..... 224, 3750, 3752  
 \MT@ex@max ..... 236, 1309, 1310  
 \MT@ex@min ..... 236, 1312, 1313  
 \MT@ex@setname ..... 2660  
 \MT@ex@split ..... 1296, 1300  
 \MT@exp@one@n .....  
 .. 313, 497, 623, 788, 790, 841, 881, 944,  
 1556, 1960, 2343, 2350, 2382, 2415, 2426,  
 2437, 2456, 2636, 2653, 2683, 2732, 3057, 3900  
 \MT@exp@two@c .....  
 .. 315, 335, 342, 354, 532, 787, 820, 822,  
 824, 835, 1670, 1703, 1706, 2120, 2125, 2208  
 \MT@exp@two@n ..... 317, 899, 908, 2484, 2590  
 \MT@expandfont ..... 1257, 1273, 1277  
 \MT@expansion ..... 802, 1246, 3779  
 \MT@expansionfalse ..... 215, 3652, 3744  
 \MT@expansiontrue ..... 215, 3461  
 \MT@extra@context ..... 2437,  
 2778, 2792, 2815, 2833, 2847, 2895, 2898,  
 2899, 2901, 2936, 2950, 2955, 3043, 3187,  
 3191, 3194, 3197, 3200, 3204, 3205, 3207, 3234  
 \MT@extra@inputenc ..... 3042, 3066, 3071, 3072  
 \MT@factor@default ..... 248, 3442, 3662  
 \MT@family ... 788, 846, 900, 943, 2030, 2724, 2725  
 \MT@familyalias ..... 789,  
 790, 852, 907, 909, 911, 913, 2040, 2042, 2726  
 \MT@feat ..... 856, 976, 1086, 1088, 1090,  
 1091, 1093, 1094, 1100, 1102, 1105, 1113–  
 1117, 1119, 1121–1125, 1128, 1129, 1134,  
 1138, 1141, 1144, 1147–1149, 1158, 1159,  
 1164, 1182, 1672, 1926, 1928, 1931, 1932,  
 1936, 1938, 1939, 1941, 2066, 2068, 2070,  
 2074, 2078, 2082, 2083, 2086, 2094, 2096,  
 2097, 2099, 2106, 2110, 2244, 2245, 2274, 2276  
 \MT@features ..... 620,  
 2437, 2450, 2636, 2647, 2677, 2920, 3052, 3486  
 \MT@features@long ..... 620, 623, 628, 2412, 3064  
 \MT@file@list ..... 1947,  
 1949, 1957, 1960, 1962, 1967, 1970, 2732, 2736  
 \MT@find@file ..... 788, 790, 1947  
 \MT@fix@font@set ..... 2503, 3657  
 \MT@font ..... 312,  
 787, 835, 842, 972, 979, 985, 986, 1017,  
 1018, 1023, 1024, 1044, 1045, 1066, 1071,  
 1077, 1172, 1175, 1208, 1209, 1211, 1212,  
 1278, 1282, 1316, 1317, 1361, 1363, 1406,  
 1407, 1413, 1414, 1420, 1421, 1432–1434,  
 1436, 1437, 1442–1444, 1503, 1504, 1510,  
 1511, 1522, 1523, 1525, 1526, 1531, 1532,  
 1556, 1560, 1604, 1900, 2278, 2314–2316,  
 2318, 2331, 2332, 2343, 2345, 2351, 2372, 2383  
 \MT@font@copy ..... 816, 817, 824, 825, 835–837  
 \MT@font@list ..... 2278, 2343–2345, 2406  
 \MT@font@orig ..... 817  
 \MT@font@sets ..... 2488, 2503, 2594, 3657  
 \MT@gdef@n ..... 303, 2771, 2785,  
 2807, 2840, 2854, 2991, 3011, 3024, 3074, 3488  
 \MT@get@axis ..... 2610–2613, 2623  
 \MT@get@basefamily ..... 1959, 1997  
 \MT@get@basefamily@ ..... 2003, 2006  
 \MT@get@char@unit ... 1008, 1136, 1162, 1396, 1493  
 \MT@get@charwd ..... 1043, 1062, 1139, 1162  
 \MT@get@config ..... 3465  
 \MT@get@ex@opt ..... 1249, 1264, 1334  
 \MT@get@ex@opt@ ..... 1342–1344, 1346, 1352  
 \MT@get@font ..... 2589, 2599  
 \MT@get@font@ ..... 2601, 2607, 2908  
 \MT@get@font@and@size ..... 2894, 2906  
 \MT@get@font@dimen ..... 1065, 1142  
 \MT@get@font@dimen@six . 959, 971, 1374, 1471, 1570  
 \MT@get@highlevel ... 2483, 2493, 2625, 2871, 3082  
 \MT@get@inh@list ..... 962, 1251, 1377, 1474, 2092  
 \MT@get@listname ..... 2012, 2068, 2094  
 \MT@get@listname@ ..... 2012  
 \MT@get@ls@basefont ..... 1583, 1693, 1700  
 \MT@get@opt ..... 960, 1111, 1375, 1472  
 \MT@get@range ..... 2512, 2522, 2881  
 \MT@get@size ..... 2529, 2534, 2543, 2562, 2621  
 \MT@get@slot ..... 1006,  
 1303, 1394, 1491, 1907, 2113, 3106, 3112  
 \MT@get@space@unit .....  
 .. 1136, 1236, 1404, 1411, 1418, 1501, 1508  
 \MT@get@tr@opt ..... 1572, 1641  
 \MT@get@tr@opt@ ..... 1659–1662, 1664  
 \MT@get@unit ..... 1144, 1152, 1652  
 \MT@get@unit@ ..... 1152

- \MT@glet . 301, 640, 820, 1703, 1833, 2312, 2407–  
 2410, 2726, 2949, 3129, 3390, 3392, 3830, 3835  
 \MT@gobble@settings . . . . . 970, 977  
 \MT@if@false . . . . . 213, 2355, 3232, 3931, 3938  
 \MT@if@list@exists . . . . .  
 . . . 958, 1248, 1263, 1373, 1470, 1571, 2065  
 \MT@if@true . . . . . 213,  
 2348, 3241, 3245, 3253, 3258, 3932–3936, 3939  
 \MT@if@defined@c@T . . . . . 321,  
 686, 687, 707, 711, 1026, 1051, 1318, 1423,  
 1513, 1600, 2195, 2724, 3071, 3176, 3498, 3501  
 \MT@if@defined@c@TF . . . . .  
 . . . . . 321, 654, 1575, 1586, 1589, 1593,  
 1691, 1760, 1873, 1904, 1905, 2069, 2095, 3186  
 \MT@if@defined@n@T . . . . .  
 . . . . . 321, 644, 878, 896, 924, 940, 1027,  
 1147, 1183, 1319, 1347, 1424, 1514, 1644,  
 1646, 1665, 2048, 2273, 2470, 2719, 2795,  
 2926, 3135, 3142, 3149, 3155, 3187, 3197, 3619  
 \MT@if@defined@n@TF . . . . . 321, 852,  
 862, 879, 897, 925, 941, 1113, 1121, 1336,  
 1353, 1894, 1931, 2036, 2121, 2662, 2666,  
 2692, 3162, 3214, 3223, 3285, 3487, 3491, 3898  
 \MT@if@dim . . . 437, 552, 553, 557, 558, 2545, 2554,  
 3239–3241, 3243, 3244, 3251–3253, 3256, 3257  
 \MT@if@dimen . . . . 402, 2574, 2980, 2988, 3000, 3451  
 \MT@if@empty . . . . . 357, 985, 986, 1015,  
 1021, 1225, 1226, 1403, 1410, 1417, 1442–  
 1444, 1456–1458, 1460–1462, 1500, 1507,  
 1531, 1532, 1548, 1549, 1741, 1742, 1752,  
 1770, 1843, 1878, 2417, 2449, 2454, 2495,  
 2501, 2523, 2524, 2537, 2538, 2600, 2615,  
 2626, 2646, 2651, 2676, 2681, 2754, 2822,  
 2907, 2917, 2922, 2936, 2948, 2955, 3009,  
 3051, 3056, 3086, 3298, 3329, 3467, 3509, 3566  
 \MT@if@int . . . . . 370, 2174, 2823, 3010, 3436, 3444  
 \MT@if@streq . . . . . 444, 1189, 1927, 2067,  
 2082, 2224, 2227, 2381, 2588, 2893, 2979,  
 2997, 2999, 3301, 3303, 3306, 3309, 3331,  
 3333, 3347, 3348, 3415, 3416, 3421, 3425,  
 3450, 3511, 3518, 3522, 3529, 3568, 3574, 3872  
 \MT@in@clist . . . . 518, 623, 645, 881, 899, 908,  
 944, 1556, 1949, 1960, 2343, 2350, 2415, 2732  
 \MT@in@rlist . . . . . 546, 927, 2057  
 \MT@in@rlist@ . . . . . 546  
 \MT@in@rlist@ . . . . . 546  
 \MT@in@tlist . . . . . 534, 2009, 2426  
 \MT@in@tlist@ . . . . . 534  
 \MT@in@not@false . . . . . 72  
 \MT@in@not@true . . . . . 72  
 \MT@increment . . . . . 582, 3134, 3141, 3148, 3154  
 \MT@info . . . . . 49, 64, 69, 2664, 3422,  
 3514, 3520, 3526, 3533, 3571, 3576, 3604, 3608  
 \MT@info@missing@char . . . . . 1047, 1074  
 \MT@info@nl . . . . . 49, 65, 66, 70, 655,  
 1076, 3413, 3423, 3474, 3492, 3495, 3655,  
 3661, 3668, 3751, 3780, 3795, 3801, 3806,  
 3810, 3816, 3820, 3851, 3857, 3863, 3908, 3919  
 \MT@inh@do . . . . . 2105, 3097  
 \MT@inh@feat . . . . . 3042, 3051, 3054, 3065  
 \MT@inh@split . . . . . 3099, 3103  
 \MT@inlist@false . . . . . 518, 522, 535, 547, 2358  
 \MT@inlist@true . . . . . 518, 524, 542, 554, 559, 2358  
 \MT@is@active . . . . . 2120, 2187  
 \MT@is@char . . . . . 2125, 2208, 2214  
 \MT@is@composite . . . . . 2123, 2235  
 \MT@is@feature . . . . . 622, 2455, 2652, 2682  
 \MT@is@letter . . . . . 2118, 2141, 2211, 2240  
 \MT@is@number . . . . . 2157, 2162  
 \MT@is@symbol . . . . . 2122, 2205  
 \MT@iterate . . . . . 568  
 \MT@kerning . . . . . 805, 1467, 3819  
 \MT@kerning@false . . . . . 221  
 \MT@kerning@true . . . . . 221  
 \MT@kn@c@name . . . . . 1476, 1478, 1539, 2847  
 \MT@kn@context . . . . . 2400, 2437  
 \MT@kn@do . . . . . 1479, 1482  
 \MT@kn@doc@contexts . . . . . 2437  
 \MT@kn@factor . . . . . 224  
 \MT@kn@factor@ . . . . . 1113  
 \MT@kn@inh@name . . . . . 1513, 1514, 1516  
 \MT@kn@max . . . . . 236  
 \MT@kn@min . . . . . 236  
 \MT@kn@setname . . . . . 2660  
 \MT@kn@split . . . . . 1484, 1488  
 \MT@kn@split@val . . . . . 1494, 1498  
 \MT@kn@unit . . . . . 224  
 \MT@kn@unit@ . . . . . 1121, 1542  
 \MT@led@kern . . . . . 651  
 \MT@led@unhbox@line . . . . . 651  
 \MT@led@mac@setup . . . . . 651, 703  
 \MT@let@cn . . . . . 310, 311, 853, 966, 1255,  
 1337, 1381, 1478, 1645, 1647, 1651, 1926,  
 1938, 2062, 2066, 2099, 2314, 2566, 2930, 3070  
 \MT@let@nc . . . . . 309,  
 650, 1149, 1735, 1867, 2078, 2106, 2110,  
 2332, 2353, 2406, 2472, 2474, 2480, 2585,  
 2800, 2977, 2981, 2986, 2989, 2995, 2998,  
 3001, 3030, 3216, 3221, 3307, 3310, 3525, 3532  
 \MT@let@nn . . . . . 311, 1114,  
 1119, 1122, 1134, 1354, 1357, 1666, 2425, 3494  
 \MT@letterspace . . . . .  
 . . . 252, 1575, 1645, 1655, 3838, 3839, 3841  
 \MT@letterspace@ . . . . . 994, 1051, 1052, 1575–  
 1577, 1580, 1584, 1588, 1597, 1605, 1614,  
 1634, 1636, 1637, 1766, 1770, 1842, 1879, 1882  
 \MT@letterspace@default . . . . . 252, 3839  
 \MT@listname . . . . .  
 2012, 2059, 2062, 2069, 2070, 2072, 2074,  
 2095, 2096, 2098, 2099, 2103, 2106, 3115, 3121  
 \MT@load@inputenc . . . . . 1190, 1193, 1202  
 \MT@load@list . . . . . 964, 1253, 1379, 1476, 1924  
 \MT@loop . . . . . 568, 576, 3220  
 \MT@lower . . . . . 2516, 2522, 2884  
 \MT@ls@adjust . . . . . 1613, 1839, 1840  
 \MT@ls@basefont . . . . . 1701–1703, 1706  
 \MT@ls@outer@k . . . . . 1615, 1619, 1625, 1782, 1818, 1887  
 \MT@ls@too@large . . . . . 1846, 1850, 3841  
 \MT@lsfont . . . . .  
 1579, 1584, 1597, 1598, 1601, 1604, 1636,  
 1637, 1670, 1678, 1713, 1715, 1729, 1737,  
 1743, 1747, 1754, 1764, 1768, 1771, 1880, 1883  
 \MT@lslig . . . . . 1690  
 \MT@lua . . . . . 285, 375, 407, 449  
 \MT@map@clist@ . . . . . 490

- $\backslash\text{MT@map@clist@c}$  ... 490, 826, 1906, 2349, 2361, 2368, 2405, 2412, 2450, 2452, 2647, 2649, 2677, 2679, 2920, 3052, 3054, 3064, 3486, 3617  
 $\backslash\text{MT@map@clist@n}$  ... 490, 861, 1893, 2437, 2481, 2510, 2586, 2636, 2709, 2769, 2869, 2879, 2891, 2993, 3007, 3109, 3293, 3296, 3325, 3327, 3356, 3431, 3507, 3564  
 $\backslash\text{MT@map@tlist@}$  ... 508  
 $\backslash\text{MT@map@tlist@c}$  ... 508, 537, 548, 829, 1028, 1320, 1425, 1515, 2016, 2374, 3188, 3233, 3657  
 $\backslash\text{MT@map@tlist@n}$  ... 508, 1988, 2216, 3128  
 $\backslash\text{MT@maybe@do}$  ... 856, 956, 1246, 1371, 1468, 1558  
 $\backslash\text{MT@maybe@etex}$  ... 257, 321, 346, 583, 1034, 1074  
 $\backslash\text{MT@maybe@etex@}$  ... 257  
 $\backslash\text{MT@maybe@rem@from@list}$  ... 2376, 2380  
 $\backslash\text{MT@MT}$  ... 2, 49, 52, 55, 57, 59, 64, 65, 67, 68, 174, 181, 292, 634, 698, 700, 2957, 3468, 3604, 3608, 3656, 3876, 3912, 3913  
 $\backslash\text{MT@next@listname}$  ... 2019, 2028  
 $\backslash\text{MT@next@listname@}$  ... 2037, 2049, 2055  
 $\backslash\text{MT@n@ligatures}$  ... 1900, 2748  
 $\backslash\text{MT@n@setname}$  ... 1892, 2748  
 $\backslash\text{MT@noligatures}$  ... 807, 1889, 3887  
 $\backslash\text{MT@noligatures@}$  ... 1713, 1715, 1900, 1903  
 $\backslash\text{MT@noligatures@false}$  ... 218  
 $\backslash\text{MT@noligatures@true}$  ... 218, 2756  
 $\backslash\text{MT@nonselected@false}$  ... 1260, 1274  
 $\backslash\text{MT@nonselected@true}$  ... 1260, 1262  
 $\backslash\text{MT@noreset@false}$  2140, 2147, 2156, 2177, 2224, 2227  
 $\backslash\text{MT@noreset@true}$  ... 2116, 2140, 2165  
 $\backslash\text{MT@old@cmd}$  ... 39, 44–47  
 $\backslash\text{MT@opt@autofalse}$  ... 3272  
 $\backslash\text{MT@opt@autot@true}$  ... 3272, 3358  
 $\backslash\text{MT@opt@def@set}$  ... 3284, 3312, 3337  
 $\backslash\text{MT@opt@DVIfalse}$  ... 3273  
 $\backslash\text{MT@opt@DVIftrue}$  ... 3273, 3361, 3364  
 $\backslash\text{MT@opt@expansion@false}$  ... 3271  
 $\backslash\text{MT@opt@expansion@true}$  ... 3271  
 $\backslash\text{MT@options}$  ... 2857  
 $\backslash\text{MT@optwarn@admissible}$  ... 3274, 3349, 3425  
 $\backslash\text{MT@optwarn@nan}$  ... 3278, 3438, 3446  
 $\backslash\text{MT@orig@add@accent}$  ... 2333  
 $\backslash\text{MT@orig@foreign@language}$  ... 3925, 3927  
 $\backslash\text{MT@orig@pickupfont}$  ... 729, 762, 767, 2283, 2335  
 $\backslash\text{MT@orig@py@macron}$  ... 759, 763, 768  
 $\backslash\text{MT@orig@select@language}$  ... 3920, 3922  
 $\backslash\text{MT@outer@kern}$  ... 1614, 1617, 1623, 1625, 1626, 1631, 1694, 1695, 1697, 1818, 1861, 1887  
 $\backslash\text{MT@outer@space}$  ... 1608, 1609, 1726, 1778, 1780, 1781, 1790, 1791, 1806, 1807, 1811, 1812  
 $\backslash\text{MT@pdf@annot}$  ... 72  
 $\backslash\text{MT@pdf@tex@no}$  ... 117, 167, 172, 277  
 $\backslash\text{MT@permute}$  2784, 2806, 2820, 2839, 2853, 3077, 3125  
 $\backslash\text{MT@permute@}$  ... 3125  
 $\backslash\text{MT@permute@@}$  ... 3125  
 $\backslash\text{MT@permute@@@}$  ... 3125  
 $\backslash\text{MT@permute@@@@}$  ... 3153, 3158  
 $\backslash\text{MT@permute@@@@@}$  ... 3166, 3168  
 $\backslash\text{MT@permute@define}$  ... 3159, 3169–3171, 3212  
 $\backslash\text{MT@permute@reset}$  ... 3128, 3218  
 $\backslash\text{MT@permute@list}$  ... 2778, 2792, 2818, 2833, 2847, 2895, 2896, 2901, 3075, 3182, 3183, 3187, 3191, 3194, 3197–3200, 3207, 3208, 3234, 3265, 3266  
 $\backslash\text{MT@pickupfont}$  ... 764, 769, 2329, 2337  
 $\backslash\text{MT@plain}$  ... 97, 162, 637, 1669, 2281, 3612, 3616  
 $\backslash\text{MT@pr@c@name}$  ... 964, 966, 1216, 2778  
 $\backslash\text{MT@pr@c@context}$  ... 2399, 2437  
 $\backslash\text{MT@pr@do}$  ... 967, 997  
 $\backslash\text{MT@pr@doc@contexts}$  ... 2437  
 $\backslash\text{MT@pr@factor}$  ... 224, 3445, 3662, 3663  
 $\backslash\text{MT@pr@factor@}$  ... 1113  
 $\backslash\text{MT@pr@inh@name}$  ... 1026, 1027, 1029  
 $\backslash\text{MT@pr@level}$  ... 224, 3660, 3661  
 $\backslash\text{MT@pr@max}$  ... 236  
 $\backslash\text{MT@pr@min}$  ... 236  
 $\backslash\text{MT@pr@setname}$  ... 2660  
 $\backslash\text{MT@pr@split}$  ... 999, 1003  
 $\backslash\text{MT@pr@split@val}$  ... 1009, 1013  
 $\backslash\text{MT@pr@unit}$  ... 224, 3452, 3664  
 $\backslash\text{MT@pr@unit@}$  ... 1121, 1219  
 $\backslash\text{MT@preset@aux}$  ... 1221, 1223, 1225, 1226, 1229, 1544, 1546, 1548, 1549  
 $\backslash\text{MT@preset@aux@factor}$  1221, 1229, 1456–1458, 1544  
 $\backslash\text{MT@preset@aux@space}$  . 1223, 1229, 1460–1462, 1546  
 $\backslash\text{MT@preset@ex}$  ... 1348, 1365  
 $\backslash\text{MT@preset@kn}$  ... 1537  
 $\backslash\text{MT@preset@kn@}$  ... 1537  
 $\backslash\text{MT@preset@pr}$  ... 1214  
 $\backslash\text{MT@preset@pr@}$  ... 1214  
 $\backslash\text{MT@preset@sp}$  ... 1449  
 $\backslash\text{MT@preset@sp@}$  ... 1449  
 $\backslash\text{MT@ProcessOptionsWithKV}$  ... 3612, 3636  
 $\backslash\text{MT@protrusion}$  ... 801, 956, 3667  
 $\backslash\text{MT@protrusion@false}$  ... 214  
 $\backslash\text{MT@protrusion@true}$  ... 214, 3458  
 $\backslash\text{MT@rbba@expansion}$  ... 615  
 $\backslash\text{MT@rbba@kerning}$  ... 615  
 $\backslash\text{MT@rbba@protrusion}$  ... 615  
 $\backslash\text{MT@rbba@spacing}$  ... 615  
 $\backslash\text{MT@rbba@tracking}$  ... 615  
 $\backslash\text{MT@register@font}$  ... 810, 2345, 2408  
 $\backslash\text{MT@register@font@cx}$  ... 2367, 2408  
 $\backslash\text{MT@register@subst@font}$  ... 2320, 2344, 2409  
 $\backslash\text{MT@register@subst@font@cx}$  ... 2360, 2409  
 $\backslash\text{MT@rem@from@clist}$  ... 529, 841, 2382, 2861  
 $\backslash\text{MT@rem@from@list}$  ... 831, 839  
 $\backslash\text{MT@repeat}$  ... 568, 578, 3226  
 $\backslash\text{MT@requires@luatex}$  ... 280, 372, 404, 446, 2940, 3379, 3789  
 $\backslash\text{MT@requires@pdftex}$  ... 276, 370, 402, 444, 651, 784, 793, 804, 806, 813, 992, 1035, 1049, 1285, 1370, 1467, 1553, 1710, 1889, 2748, 2938, 3023, 3029, 3377, 3460, 3561, 3707, 3730, 3783, 3867, 3884  
 $\backslash\text{MT@res@a}$  ... 467, 469, 520, 527, 530, 532, 536, 541  
 $\backslash\text{MT@res@b}$  ... 468, 469, 530–532, 540, 541  
 $\backslash\text{MT@reset@context}$  ... 2390, 2393, 2397, 2424  
 $\backslash\text{MT@reset@context@}$  ... 2397, 2424  
 $\backslash\text{MT@reset@ef@codes}$  ... 1250, 1272, 1285, 1349  
 $\backslash\text{MT@reset@ef@codes@}$  ... 1280, 1288, 1292  
 $\backslash\text{MT@reset@kn@codes}$  ... 1473, 1480, 1528  
 $\backslash\text{MT@reset@kn@codes@}$  ... 1528  
 $\backslash\text{MT@reset@pr@codes}$  ... 961, 968, 989  
 $\backslash\text{MT@reset@pr@codes@}$  ... 989

- \MT@reset@sp@codes ..... 1376, 1383, 1439  
 \MT@reset@sp@codes@ ..... 1439  
 \MT@restore@catcodes ..... 200, 208  
 \MT@restore@p@h ..... 684, 690, 691, 726, 739  
 \MT@saved@setupfont ..... 3602, 3609  
 \MT@scale ..... 590, 1038, 1087, 1101, 1104, 1307  
 \MT@scale@factor .... 1040, 1059, 1085, 1231, 1367  
 \MT@scale@to@em .. 1016, 1022, 1034, 1056, 1237,  
     1405, 1412, 1419, 1502, 1509, 1654, 1761, 1874  
 \MT@selectedfalse ..... 217  
 \MT@selectedtrue ..... 217  
 \MT@series ..... 846, 943, 2031, 2043  
 \MT@set@all@ex ..... 1280, 1368  
 \MT@set@all@kn ..... 1528, 1550  
 \MT@set@all@pr ..... 982, 989, 1227  
 \MT@set@all@sp ..... 1439, 1464  
 \MT@set@babel@context ..... 3897, 3923, 3928, 3942  
 \MT@set@curr@ls ..... 1605  
 \MT@set@curr@ok ..... 1617, 1618, 1631, 1795  
 \MT@set@curr@os ..... 1607, 1796  
 \MT@set@default@set ..... 2677, 2683, 2690  
 \MT@set@ex@codes ..... 1276, 3735, 3737  
 \MT@set@ex@codes@n ..... 1260, 1276, 3737  
 \MT@set@ex@codes@s ..... 1247, 3735  
 \MT@set@ex@heirs ..... 1322, 1360  
 \MT@set@inh@list ..... 3048, 3050  
 \MT@set@inputenc 963, 1180, 1252, 1378, 1475, 2104  
 \MT@set@inputenc@ ..... 1183, 1185  
 \MT@set@kn@codes ..... 1469  
 \MT@set@kn@heirs ..... 1517, 1521  
 \MT@set@listname .....  
     965, 1112, 1254, 1335, 1380, 1477, 1643, 2243  
 \MT@set@lsfont ..... 1603, 1669, 1678  
 \MT@set@named@keys 2780, 2794, 2816, 2835, 2849, 2857  
 \MT@set@pr@codes ..... 957  
 \MT@set@pr@heirs ..... 1030, 1207  
 \MT@set@sp@codes ..... 1372  
 \MT@set@sp@heirs ..... 1427, 1431  
 \MT@set@tr@codes ..... 1567, 1673, 1683  
 \MT@SetExpansion ..... 2790, 2792  
 \MT@SetExtraKerning ..... 2845, 2847  
 \MT@SetExtraSpacing ..... 2831, 2833  
 \MT@SetProtrusion ..... 2776, 2778  
 \MT@SetTracking ..... 2812, 2814  
 \MT@setup@ ..... 636, 639, 640, 3371, 3372  
 \MT@setup@contexts ..... 2389, 2404  
 \MT@setupfont .. 781, 2323, 3602, 3605, 3609, 3644  
 \MT@setupfont@hook .....  
     .. 685, 704, 706, 710, 715, 726, 739, 746, 782  
 \MT@shape ..... 846, 943, 2032, 2044  
 \MT@shorthandoff ..... 3906, 3937, 3940  
 \MT@show@pdfannot ..... 78  
 \MT@shrink ..... 224, 1267, 3674,  
     3676, 3678, 3682, 3683, 3686, 3690, 3740, 3753  
 \MT@shrink@ ..... 1267, 1278, 1334  
 \MT@shrink@default ..... 249, 3678  
 \MT@size ..... 553, 557, 558, 846, 929, 932, 943  
 \MT@size@name ..... 546, 2059  
 \MT@sp@c@name ..... 1379, 1381, 1451, 2833  
 \MT@sp@context ..... 2400, 2437, 3872  
 \MT@sp@do ..... 1382, 1385  
 \MT@sp@doc@contexts ..... 2437  
 \MT@sp@factor ..... 224  
 \MT@sp@factor@ ..... 1113  
 \MT@sp@inh@name ..... 1423, 1424, 1426  
 \MT@sp@max ..... 236  
 \MT@sp@min ..... 236  
 \MT@sp@setname ..... 2660  
 \MT@sp@split ..... 1387, 1391  
 \MT@sp@split@val ..... 1397, 1401  
 \MT@sp@unit ..... 224  
 \MT@sp@unit@ ..... 1121, 1454  
 \MT@spacing ..... 805, 1370, 3809  
 \MT@spacingfalse ..... 220  
 \MT@spacingtrue ..... 220  
 \MT@split@name ..... 787, 846  
 \MT@step ..... 224,  
     1268, 3681, 3697, 3704, 3754, 3759, 3760, 3764  
 \MT@step@ ..... 1268, 1278, 1334  
 \MT@step@default ..... 249  
 \MT@stretch ..... 224, 1266, 3671, 3672,  
     3675, 3676, 3682, 3684, 3689, 3692, 3739, 3753  
 \MT@stretch@ ..... 1266, 1278, 1334  
 \MT@stretch@default ..... 249, 3672  
 \MT@temp ..... 984–987,  
     1441–1445, 1530–1533, 1731, 1735, 1738,  
     1743, 1746, 1751, 1790, 1797, 1800, 1803,  
     1806, 1811, 1818, 1821, 1864, 1867, 1884,  
     2609, 2634, 3615, 3620, 3625, 3626, 3629, 3633  
 \MT@tempencoding ..... 3161, 3162, 3172, 3179  
 \MT@tempfamily ..... 3173  
 \MT@tempseries ..... 3174  
 \MT@tempshape ..... 3175  
 \MT@tempsize ..... 3129, 3176, 3186, 3188, 3192  
 \MT@test@ast ..... 1759, 1872, 2494, 2499  
 \MT@textls ..... 23, 1839, 1840, 1842  
 \MT@the@pr@code ..... 991, 1017, 1023, 1638  
 \MT@the@pr@code@tr ..... 991, 1638  
 \MT@tlist@break ..... 508, 543, 565, 2021, 3268  
 \MT@toks ..... 212, 1076, 1107, 1330,  
     2117, 2145, 2152, 2199, 2229, 2248, 2260, 2268  
 \MT@tr@c@name .....  
     ... 1644–1647, 1651, 1665, 1666, 2817, 2824  
 \MT@tr@context ..... 2400, 2437  
 \MT@tr@doc@contexts ..... 2437  
 \MT@tr@factor@ ..... 1653  
 \MT@tr@font@list ..... 1554  
 \MT@tr@ispace ..... 1586, 1587, 1659  
 \MT@tr@ligatures ..... 1600, 1662, 1712, 1715  
 \MT@tr@max ..... 236, 1851–1853  
 \MT@tr@min ..... 236, 1855–1857  
 \MT@tr@noligatures ..... 1600, 1709  
 \MT@tr@okern ..... 1593, 1661  
 \MT@tr@ospace ..... 1589, 1590, 1659  
 \MT@tr@outer@icr ..... 1800, 1823  
 \MT@tr@outer@icr@ ..... 1823  
 \MT@tr@outer@l ..... 1611, 1775  
 \MT@tr@outer@next ..... 1786, 1825  
 \MT@tr@outer@r ..... 1633, 1786, 1823  
 \MT@tr@outer@r@ ..... 1786  
 \MT@tr@set@okern ..... 1594, 1861  
 \MT@tr@set@okern@ ..... 1865, 1866, 1871  
 \MT@tr@set@space ..... 1592, 1727  
 \MT@tr@set@space@ ..... 1732–1734, 1740  
 \MT@tr@set@space@@ ..... 1745, 1750, 1753, 1758  
 \MT@tr@setname ..... 2660

- `\MT@tr@unit@` ..... 1646, 1760, 1873  
`\MT@tracking` .....  
794, 1554, 1673, 2325, 3588, 3589, 3791, 3800  
`\MT@tracking@` ..... 1554, 3589  
`\MT@trackingfalse` ..... 222  
`\MT@trackingtrue` ..... 222  
`\MT@try@order` ..... 2016, 2024  
`\MT@undefined@char` ..... 2192, 2194, 2204  
`\MT@upper` ..... 2516, 2522, 2884  
`\MT@use@set` ..... 2647, 2653, 2660  
`\MT@val` ..... 2417–2419, 2421, 2427, 2429,  
2432, 2482, 2486, 2494, 2496, 2511–2513,  
2525, 2528, 2530, 2533, 2535, 2536, 2542,  
2544, 2545, 2547, 2550, 2552, 2563, 2566,  
2570, 2574–2576, 2578, 2580, 2587–2589,  
2592, 2602–2604, 2617, 2619, 2624, 2626,  
2628, 2630, 2870, 2872, 2880–2882, 2892–  
2894, 2897, 2902, 3080, 3081, 3083, 3087,  
3108, 3115, 3120, 3121, 3285, 3286, 3289,  
3297, 3298, 3301, 3303, 3306, 3309, 3328,  
3329, 3331, 3333, 3508, 3509, 3511, 3518,  
3522, 3529, 3536, 3565, 3566, 3568, 3574, 3577  
`\MT@variants` ..... 2009, 2702, 2712  
`\MT@vinfo` ..... 49, 66, 800, 874, 1116, 1125,  
1129, 1164, 1338, 1355, 1569, 1577, 1911,  
1920, 1932, 1956, 1966, 1969, 2072, 2074,  
2080, 2398, 2421, 2734, 2740, 3413, 3420, 3899  
`\MT@warn@ascii` ..... 2154, 2247  
`\MT@warn@axis@empty` ..... 2616, 2627, 2632  
`\MT@warn@code@too@large` ..... 1091, 1094, 1098  
`\MT@warn@err` ..... 49, 3417, 3418  
`\MT@warn@ex@too@large` ..... 1310, 1313, 1328  
`\MT@warn@maybe@inputenc` ..... 2261, 2269, 2272  
`\MT@warn@nodim` ..... 2972, 2982, 2990, 3002  
`\MT@warn@number@too@large` ..... 2181, 2253  
`\MT@warn@preset@t@width` ... 1220, 1240, 1455, 1543  
`\MT@warn@rest` ..... 2134, 2258  
`\MT@warn@tracking@DVI` ..... 1578, 3823  
`\MT@warn@unknown` ..... 2131, 2266  
`\MT@warning` ..... 40,  
49, 67, 1830, 1852, 1856, 1914, 2083, 2471,  
2578, 2633, 2720, 2742, 2763, 2825, 2927,  
2973, 3013, 3026, 3088, 3180, 3198, 3265,  
3417, 3422, 3498, 3535, 3554, 3577, 3594–3596  
`\MT@warning@nl` ..... 49, 61, 68, 171, 291, 633,  
667, 676, 717, 973, 1067, 1105, 1157, 1197,  
1241, 1329, 1720, 2248, 2254, 2259, 2267,  
2296, 2797, 3275, 3280, 3289, 3418, 3423,  
3453, 3481, 3641, 3699, 3710, 3721, 3741,  
3762, 3790, 3825, 3848, 3854, 3860, 3873, 3944  
`\MT@while@num` ..... 574, 580, 742–744  
`\MT@with@babel@and@T` .....  
... 643, 686, 687, 705, 709, 3932–3936, 3939  
`\MT@with@package@T` ..... 641, 688, 691,  
696, 703, 713, 724, 738, 740, 753, 758, 2307  
`\MT@xadd` ..... 76, 476, 1957, 1962, 1967,  
1970, 2362, 2370, 2429, 2514, 2736, 2883, 3114  
`\MT@xaddb` ..... 483, 2900, 3190  
`\MT@xdef@n` .. 305, 2663, 2667, 2694, 2699, 2723,  
2824, 2895, 2918, 3038, 3072, 3207, 3286, 3288
- N**
- `\normalfont` ..... 780, 3772, 3776
- `\nullfont` ..... 780
- P**
- `\pdfadjustinterwordglue` ..... 3590, 3805  
`\pdfadjustspacing` ..... 3560, 3750, 3775  
`\pdfappendkern` ..... 3592, 3815  
`\pdfcopyfont` ..... 824  
`pdfcpot (package)` ..... 26, 49, 58, 86, 178  
`\pdffontexpand` ..... 1278  
`\pdfnoligatures` ..... 1913, 1919  
`\pdfprependkern` ..... 3591, 3814  
`\pdfprotrudechars` ..... 3559, 3660, 3797, 3834  
`\pdftracingfonts` ..... 279  
`\pickup@font` ..... 729, 762, 764,  
767, 769, 2295, 2297, 2306, 2329, 2335, 2337  
`pifont (package)` ..... 87  
`pinyin (package)` ..... 50, 183  
`protrusion (option)` ..... 6, 7, 11, 110  
`PSNFSS (package)` ..... 174  
`pstricks (package)` ..... 9, 118  
`pxfonts (package)` ..... 21, 126
- Q**
- `qfonts (package)` ..... 21, 126, 181
- R**
- `relsize (package)` ..... 94  
`\rightmargin` ..... 660  
`\rmode` ..... 986, 1023, 1024, 1209, 1212, 1637
- S**
- `selected (option)` ..... 8, 14, 31, 62, 80, 111, 179  
`\selectfont` ..... 1674, 2392, 2402, 3950  
`\set@fontsize` ..... 2570  
`\SetExpansion` ..... 10, 14, 2788  
`\SetExtraKerning` ..... 12, 19, 2843  
`\SetExtraSpacing` ..... 13, 18, 2829  
`\SetProtrusion` ..... 9, 12, 2774  
`\SetTracking` ..... 11, 16, 2810  
`\sfcode` ..... 3871  
`\shbcode` ..... 1420, 1421, 1434, 1437, 1444  
`\showhyphens` ..... 3770, 3774  
`shrink (option)` .. 8, 15, 31, 113, 119, 120, 181, 183  
`slantsc (package)` ..... 152  
`soul (package)` ..... 50, 74, 182  
`spacing (option)` ..... 7, 11, 29, 110  
`\stbcode` ..... 1413, 1414, 1433, 1437, 1443  
`step (option)` ... 8, 31, 113, 119, 120, 179, 181, 183  
`stretch (option)` ... 8, 15, 25, 31, 113, 118–120, 183
- T**
- `\tagcode` ..... 1601, 1905, 1909  
`tex4ht (package)` ..... 49, 182  
`\textls` ..... 23, 26, 27, 734, 755, 1686, 1837  
`\textls*` ..... 23  
`\textmicrotypecontext` ..... 22, 22, 730, 2396  
`tipa (package)` ..... 87  
`trace (package)` ..... 87, 182  
`\tracingmicrotype` ..... 60  
`\tracingmicrotypeinpdf` ..... 71  
`\tracingmicrotypeinpdfallfalse` ..... 77  
`\tracingmicrotypeinpdfalltrue` ..... 77  
`tracking (option)` ..... 7, 11, 23, 29, 110  
`txfonts (package)` ..... 21, 126

U		W	
ucs (package) . . . . .	84, 184	wordcount (package) . . . . .	47, 183
ulgothic (package) . . . . .	21		
unit (option) . . . . .	7, 30, 180–182	Y	
\UseMicrotypeSet . . . . .	7, 11, 34, 46, 2465, 2644	yfonts (package) . . . . .	24
V		Z	
verbose (option) . . . . .	9, 29, 31, 112, 180, 184	zefonts (package) . . . . .	21, 126

## 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 ‘cfigguide.tex’ and ‘modguide.tex’ in the base L<sup>A</sup>T<sub>E</sub>X distribution for suggestions.

### Definitions

In this license document the following terms are used:

**Work:** Any work being distributed under this License.

**Derived Work:** Any work that under any applicable law is derived from the Work.

**Modification:** Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

**Modify:** To apply any procedure that produces a Derived Work under any applicable law.

**Distribution:** Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun’s Network File System (NFS).

**Compiled Work:** A version of the Work that has been processed into a form where it is directly usable on a

computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

**Current Maintainer:** A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the ‘Copyright Holder’ under any applicable law.

**Base Interpreter:** A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a ‘L<sup>A</sup>T<sub>E</sub>X-Format’ or in the case of files belonging to the ‘L<sup>A</sup>T<sub>E</sub>X-format’ a program implementing the ‘T<sub>E</sub>X language’.



## Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.
2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.
5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
  - (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
  - (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.
  - (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
- (d) You distribute at least one of the following with the Derived Work:
  - i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
  - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.
10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
  - (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.
11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

## No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any au-

thor named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

## Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
  - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
  - (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current

Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a L<sup>A</sup>T<sub>E</sub>X work, this could be done, for example, by posting to `comp.text.tex`.)

3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.
- (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.
4. If you make an ‘intention announcement’ as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.
5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

## Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

### Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not

refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document ‘modguide.tex’ in the base L<sup>A</sup>T<sub>E</sub>X distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing L<sup>A</sup>T<sub>E</sub>X under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L<sup>A</sup>T<sub>E</sub>X, the discussion in ‘modguide.tex’ may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

## A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

## How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
%   http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
```

```
% This work has the LPPL maintenance status 'maintained'.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the ‘Work’ referring to the three files ‘pig.dtx’, ‘pig.ins’, and ‘pig.sty’ (the last being generated from ‘pig.dtx’ using ‘pig.ins’), the ‘Base Interpreter’ referring to any ‘L<sup>A</sup>T<sub>E</sub>X-Format’, and both ‘Copyright Holder’ and ‘Current Maintainer’ referring to the person ‘M. Y. Name’.

If you do not want the Maintenance section of LPPL to apply to your Work, change ‘maintained’ above into ‘author-maintained’. However, we recommend that you use ‘maintained’ as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

## Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case (e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

## Important Recommendations

### Defining What Constitutes the Work

The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.