

# The microtype package

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

R Schlicht  
w.m.l@gmx.net

v2.3a  
2008/02/29

---

## 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 `miniltx`, provides the user commands for letterspacing only, omitting support for all other extensions (see section 7).

This package is copyright © 2004–2008 R Schlicht. It may be distributed and/or modified under the conditions of the  [\$\LaTeX\$  Project Public License](#), either version 1.3c of this license or (at your option) any later version. This work has the LPPL maintenance status ‘author-maintained’.

---

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

## Contents

<b>1</b>	<b>Micro-Typography with pdf<math>\TeX</math></b>	<b>4</b>
<b>2</b>	<b>Getting Started</b>	<b>5</b>
<b>3</b>	<b>Options</b>	<b>6</b>
3.1	Micro-Typographic Options . . . . .	6
3.2	Options for Character Protrusion . . . . .	7
3.3	Options for Font Expansion . . . . .	8
3.4	Option for Tracking/Letterspacing . . . . .	9
3.5	Miscellaneous Options . . . . .	9
3.6	Changing Options Later . . . . .	10
<b>4</b>	<b>Selecting Fonts for Micro-Typography</b>	<b>10</b>
<b>5</b>	<b>Micro Fine Tuning</b>	<b>12</b>
5.1	Character Protrusion . . . . .	13
5.2	Font Expansion . . . . .	14
5.3	Tracking . . . . .	16
5.4	Interword Spacing . . . . .	18
5.5	Additional Kerning . . . . .	19
5.6	Character Inheritance . . . . .	20
5.7	Configuration Files . . . . .	20
<b>6</b>	<b>Context-sensitive Setup</b>	<b>22</b>
<b>7</b>	<b>Letterspacing revisited</b>	<b>23</b>
<b>8</b>	<b>Disabling Ligatures</b>	<b>25</b>
<b>9</b>	<b>Hints and Caveats</b>	<b>25</b>
<b>10</b>	<b>Contributions</b>	<b>28</b>
<b>11</b>	<b>Acknowledgments</b>	<b>28</b>
<b>12</b>	<b>References</b>	<b>29</b>
<b>13</b>	<b>Short History</b>	<b>29</b>
<b>14</b>	<b>Implementation</b>	<b>33</b>
14.1	Preliminaries . . . . .	34
	[Debugging – 35] [Requirements – 37] [Declarations – 39] [Auxiliary Macros – 40] [Compatibility – 48]	
14.2	Font Setup . . . . .	52
	[Protrusion – 57] [Expansion – 63] [Interword Spacing (Glue) – 66] [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 – 99] [Character Inheritance – 104] [Permutation – 106]	
14.4	Package Options . . . . .	110
	[Declaring the Options – 110] [Reading the Configuration File – 114]	
	[Hook for Other Packages – 115] [Changing Options Later – 115]	
	[Processing the Options – 117]	
<b>15</b>	<b>Configuration Files</b>	<b>125</b>
15.1	Font Sets . . . . .	125
15.2	Font Variants and Aliases . . . . .	126
15.3	Interaction with babel . . . . .	127
15.4	Note on Admissible Characters . . . . .	127
15.5	Character Inheritance . . . . .	128
	[OT1 – 128] [T1 – 128] [LY1 – 129] [OT4 – 129] [QX – 130] [T5 – 131]	
	[Euro symbols – 131]	
15.6	Tracking . . . . .	132
15.7	Font Expansion . . . . .	132
15.8	Character Protrusion . . . . .	133
	[Default – 134] [Italics – 141] [Small Caps – 150] [Italic Small Caps – 152]	
	[Text Companion – 154] [Computer Modern math – 159] [AMS symbols – 163] [Euler – 166] [Euro symbols – 171]	
15.9	Interword Spacing . . . . .	172
	[Nonfrenchspacing – 173]	
15.10	Additional Kerning . . . . .	174
	[French – 175] [Turkish – 175]	
<b>16</b>	<b>Auxiliary File for Micro Fine Tuning</b>	<b>176</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	Availability of micro-typographic features . . . . .	7
2	Predefined font sets . . . . .	12
3	Fonts with tailored protrusion settings . . . . .	21
4	Order for matching font attributes . . . . .	79

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

pdf $\text{\TeX}$ , the  $\text{\TeX}$  extension written by Hàn Thế Thành, introduces a number of micro-typographic features that make it the tool of choice not only for the creation of electronic documents but also of works of outstanding time-honoured typography: most prominently, *character protrusion* (also known as margin kerning) and *font expansion*. Quoting Hàn Thế Thành's thesis:

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

Protrusion	off
Expansion	off

Both features are enabled throughout this document.

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

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

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

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

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

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

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

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

*Adjustment of interword spacing* is based upon the idea that in order to achieve a uniform greyness of the text, the space between words should also depend on the surrounding characters. For example, if a word ends with an ‘r’, the following space should be a tiny bit smaller than that following, say, an ‘m’. You can think of this concept as an extension to T<sub>E</sub>X’s ‘space factors’. However, while space factors will influence all three parameters of interword space (or glue) by the same amount – the kerning, the maximum amount that the space may be stretched and the maximum amount that it may be shrunk – pdfT<sub>E</sub>X provides the possibility to modify these parameters independently from one another. Furthermore, the values may be set differently for each font. And, probably most importantly, the parameters may not only be increased but also decreased. This feature may enhance the appearance of paragraphs even more. Emphasis in the last sentence is on the word ‘may’: this extension is still highly experimental – in particular, only ending characters will currently have an influence on the interword space. Also, the settings that are shipped with microtype are but a first approximation, and I would welcome corrections and improvements very much. I suggest reading the reasoning behind the settings in section 15.9.

Setting *additional kerning* for characters of a font is especially useful for languages whose typographical tradition requires certain characters to be separated by a space. For example, it is customary in French typography to add a small space before question mark, exclamation mark and semi-colon, and a bigger space before the colon and the guillemets. Until now, this could only be achieved by making these characters active (for example by the `babel` package), which may not always be a robust solution. In contrast to the standard kerning that is built into the fonts (which will of course apply as usual), this additional kerning is based on single characters, not on character pairs.

The possibility, finally, to *disable all ligatures* of a font may be useful for type-writer fonts.

The microtype package provides an interface to all these micro-typographic extensions. All micro-typographic aspects may be customised to your taste and needs in a straight-forward manner. The next chapters will present a survey of all options and customisation possibilities.

## 2 Getting Started

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

This will be sufficient in most cases, and if you are not interested in fine-tuning the micro-typographic appearance of your document (which would seem unlikely, since using this package is proof of your interest in typographic issues), you may

actually skip the rest of this document. If this, on the other hand, does not satisfy you – be it for theoretical or practical reasons – this manual will guide you on the path to the desired results along the following milestones:

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

## 3 Options

Like many other L<sup>A</sup>T<sub>E</sub>X packages, the `microtype` package accepts options in the well known key=value syntax. In the following, you’ll find a description of all **keys** and their possible values (‘true’ may be omitted; multiple values, where allowed, must be enclosed in braces; the default value is shown on the right, preceded by an asterisk if it is contingent on the 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}
```

Table 1: Availability of micro-typographic features

pdfTeX		Micro-typographic features					
version	output	Protrusion	Expansion	(= auto)	Kerning	Spacing	Tracking
< 0.14f	DVI/PDF	∅	∅	∅	∅	∅	∅
≥ 0.14f	DVI/PDF	★	☒	∅	∅	∅	∅
≥ 1.20	DVI	★	☒	∅	∅	∅	∅
	PDF	★	★	★	∅	∅	∅
≥ 1.40	DVI	★	☒	∅	☒	☒	∅
	PDF	★	★	★	☒	☒	☒
★ = enabled   ☒ = not enabled   ∅ = not available							

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

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

tracking

spacing

kerning

true, false, *<font set name>*

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.

false

In table 1, you find an overview of which micro-typographic features are available and enabled by default for the relevant pdfTeX versions and the output mode.

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>*

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).

1000

**unit** character,  $\langle dimension \rangle$  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  $\langle font\ name \rangle \pm \langle expansion\ value \rangle$ , e. g., `cmr12+10`, as described in the [pdfTeX manual](#)).

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

**stretch**  $\langle integer \rangle$  20

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

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

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

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

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

```
\usepackage{microtype}
```

**selected** true, false false

When applying font expansion, it is possible to restrict the expansion of some characters that are more sensitive to deformation than others (e. g., the 'O', in contrast to the 'I'). This is called *selected expansion*, and its usage allows to increase the stretch and shrink limits (to, say, 30 instead of 20); however, the gain is limited since at the same time the average stretch variance will be decreased.

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.

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



### 3.4 Option for Tracking/Letterspacing

**letterspace** *<integer>* 100

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

### 3.5 Miscellaneous Options

**DVIoutput** true, false \*false

pdf $\TeX$  is not only able to generate PDF output but can also spit out DVI files.<sup>3</sup> The latter can be ordered with the option `DVIoutput`, which will set `\pdfoutput` to zero.

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

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

**draft** true, false false

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

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

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

**babel** true, false false

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

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

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

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

### 3.6 Changing Options Later

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

Inside the preamble, this command accepts all package options described above (except for `config`). In the document body, this command may be used to change the general settings of the micro-typographic extensions. It then accepts 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  $\langle value \rangle default$ , e. g.,  $\rm default$ . A single asterisk means  $\langle attribute \rangle default$ , e. g.,  $\encoding default$ , respectively  $\normal size$  for the size axis. Sizes may either be specified as a dimension (‘10’ or ‘10pt’), or as a size selection command *without* the backslash. You may also specify ranges (e. g., ‘small-Large’); while the lower boundary is included in the range, the upper boundary is not. Thus, ‘12-16’ would match 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 =  $\langle encoding \rangle / \langle family \rangle / \langle series \rangle / \langle shape \rangle / \langle size \rangle$ ’. This allows you to add fonts to the set that are otherwise disjunct from it. For instance, if you wanted to have the roman family in all sizes protruded, but only the normal sized, possibly italic, typewriter font (in contrast to, say, the small one), this is how you could declare the set:

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

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

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

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

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

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

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

If a feature is enabled but no font set has been chosen explicitly, the sets declared

Table 2: Predefined font sets

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

by this command will be activated. By default, the ‘`alltext`’ font set will be used for character protrusion and additional kerning, the ‘`basictext`’ set for font expansion and interword spacing, and the ‘`smallcaps`’ set for tracking.

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

## 5 Micro Fine Tuning

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

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

The set of fonts to which the settings should apply is declared using the same syntax of  $\langle font\ axis \rangle = \langle value\ list \rangle$  pairs as for the command `\DeclareMicrotypeSet` (see section 4). The only difference is that asterisked values will be expanded immediately instead of at the end of the preamble.

To find the matching settings for a given font the package will try all combinations of font encoding, family, series, shape and size, with decreasing significance in this order. For instance, if both settings for the current family (say, T1/`cmr`///`)` and settings for italic fonts in the normal weight (T1//`m`/`it`/) exist, those for the Computer Modern Roman font would apply.<sup>4</sup> The encoding must always match.

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

## 5.1 Character Protrusion

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

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

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

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

The *protrusion settings* consist of *<character> = <protrusion factors>* pairs.

The characters may be specified either as a single character (‘A’), as a text symbol command (‘\textquoteleft’), or as a slot number: three digits for decimal notation, prefixed with ‘#’ for hexadecimal, with ‘o’ for octal (e. g., the ‘fl’ ligature in T1 encoding: 029, #1D, o35). 8-bit (and even UTF-8) characters may be entered directly or in L<sup>A</sup>T<sub>E</sub>X’s traditional 7-bit notation: both ‘\’A and ‘A are valid, provided the character is actually declared in both the input and the font encoding. 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.

<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.

For example, if you set the expansion factor for the character ‘O’ to 500, it will only be expanded or shrunk by one half of the amount that the rest of the characters will be expanded or shrunk. While the default value for character protrusion is 0 – that is, if you didn’t specify any characters, none would be protruded – the default value for expansion is 1000, which means that all characters would be expanded by the same amount.

*Options:*

**name, load, preset, inputenc, context** Analogous to `\SetProtrusion`, the optional argument may be used to assign a name to the list, to load another list, to preset all expansion factors, to set the input encoding, or to determine the context of the list (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 54). 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 T<sub>E</sub>X for a long time – is the adjustment of tracking, i. e., the uniform addition or subtraction of letter space to/from all the characters in a font. For example, it is good typographic practice to slightly space out text set in all capitals or small capitals (as in this document). Legibility may also be improved by minimally increasing the tracking of smaller and decreasing that of larger type.<sup>6</sup> The `\SetTracking` command allows to specify the tracking amount for different fonts or font sets. It will also be evaluated by the `\textls` command, which may be used for letterspacing shorter pieces of text (see section 7).

The *tracking amount* is specified in thousandths of 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

<sup>6</sup> With full-featured fonts like Computer Modern, this is usually not necessary, though, since they come in optical sizes, and the tracking of the small-capitals font is already adjusted.



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

**no ligatures** As far as pdf $\TeX$  is concerned, ligatures in letterspaced fonts would be constructed as usual, which may be advisable when changing the tracking by only a small amount. For larger letterspacing amounts, on the other hand, the normal letter space within ligatures would have displeasing effects. This key expects a comma-separated list of characters for which ligatures should be disabled; the character that begins a ligature must be specified. If the key is given without a value, *all* ligatures of the font will be disabled. This is not recommended, however, since it also entails that kerning will be switched off.<sup>7</sup> The default settings disable ligatures for the character ‘f’ only, i. e., ‘ff’, ‘fi’, ‘ffi’, etc.<sup>8</sup> In exceptional situations, you can manually break up a ligature by inserting ‘{\kern0pt}’ resp. babel’s “| shortcut, or protect it by enclosing it in \lslig (see section 7).

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

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

and then write:

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

this is the (typographically dubious) outcome:

Stop stealing sheep!

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

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

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

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

As another 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:

```
\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  $\text{\fontdimen 4}$ ). 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⟩*}

With this command, you can fine tune the extra kerning. In contrast to standard kerning, which is always associated with a *pair* of characters, and to tracking, which specifies the space between *all* characters of a font, the extra kerning relates to single characters, that is, whenever a particular character appears in the text, the specified kerning will be inserted, regardless of which character precedes resp. follows it.

I should not neglect to mention a limitation of this additional kerning: words *immediately following* such a kern (not separated by a space) will not be hyphenated, unless you insert the breakpoints manually, e. g., for kerning after the apostrophe, ‘l 'apostrophe’. This restriction of pdfTeX will hopefully be lifted soon.

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

*Options:*

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

**unit** Admissible values are: *space*, *character* and a *⟨dimension⟩*. By default, the values denote thousandths of 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, a thin space will be inserted in front of each question mark, exclamation mark and semicolon; a normal space in front of the colon. Read section 6 to learn how to activate these settings! This paragraph was input like this :

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

## 5.6 Character Inheritance

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

In most cases, accented characters should inherit the settings from the respective base character. For example, all of the characters 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

Table 3: Fonts with tailored protrusion settings

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

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

be loaded from the file `microtype.cfg`. You may extend this file with custom settings (or load a different configuration file with the ‘`config`’ option, see section 3.5).

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

`\DeclareMicrotypeVariants` {⟨list of suffixes⟩}

`\DeclareMicrotypeVariants*` On its search for a configuration file, the package will also try to remove from the font name a suffix of one or more letters that denotes a ‘variant’ of the base font (cf. [Fontname](#)). This allows it to put settings for, e. g., the fonts `padx` (expert set), `padj` (oldstyle numerals) and `pad` (plain) into one and the same file `mt-pad.cfg`. This command expects a comma-separated list of variant suffixes. The starred version appends the suffix(es) to the existing list. The default declaration 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}}}
```

<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.

```
\renewcommand*{\@footnotemark}{%
  \leavevmode \ifhmode\edef\x@sf{\the\spacefactor}\nobreak\fi
  \new@makefnmark \ifhmode\spacefactor\x@sf\fi \relax}
```

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

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

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

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

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

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

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

Instead of adding these commands manually to your document, you may also load microtype with the `babel` option. 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  
`\lsstyle` may also want to letterspace shorter pieces of text, regardless of the font in which

they are typeset.<sup>10</sup> For such ad-hoc letterspacing, microtype introduces two commands that can be used in the same way as L<sup>A</sup>T<sub>E</sub>X's text commands (independently of whether the tracking option is enabled): `\textls` expects the text in the mandatory argument, while `\lsstyle` will switch on letterspacing for all subsequent fonts until the end of the current group. The starred version of `\textls` does not add any extra kerning before or after the text, which may be useful, e. g., for section titles. By default, each character will be spaced out by  $100/1000\text{em} = 0.1\text{em}$ ; this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option, with decreasing significance in this order.

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

Since the commands `\textls` and `\lsstyle` will also evaluate the ‘no ligatures’ key for the respective font, you 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, ‘`Ausſichtsloſigkeit`’).

```
\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 L<sup>A</sup>T<sub>E</sub>X, 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).

<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.



## 8 Disabling Ligatures

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

While completely disabling all ligatures of a font (which will also switch off kerning for this font), purposely *lowers* the micro-typographic quality instead of raising it, it is especially useful for typewriter fonts, so that, e. g., in a T1 encoded font, ‘\texttt{--}’ will indeed be printed as ‘--’, not as ‘-’. `\DisableLigatures` may be used to specify, in the usual way, a set of fonts for which ligatures should be disabled, for example, of the typewriter font in T1 encoding:

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

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

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

The character that begins the ligature(s) is what matters. This command may only be used in the preamble, and only once. It requires pdf<sub>T</sub><sub>E</sub>X 1.30 or later.

## 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 pdf<sub>T</sub><sub>E</sub>X versions).* With pdf<sub>T</sub><sub>E</sub>X 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 pdf<sub>T</sub><sub>E</sub>X 1.40, which uses a different technique of expansion, the file size increase can be neglected.

*Settings for Cyrillic/Greek/Thai etc. encodings are not yet included.* The default sets of fonts for which the micro-typographic features will be enabled (see table 2) only contain those encodings for which configurations exist. Therefore, if you are using any other encoding (e. g., T2A, LGR etc.), microtype will not apply to these fonts. You have to define and activate a new font set including the encoding(s) you are using (for details, see section 4). For protrusion, 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 and interaction with other packages:* The microtype package is supposed to work happily together with all other L<sup>A</sup>T<sub>E</sub>X packages (except for pdfcp<sub>rot</sub>). However, life isn’t perfect, so problems are to be expected. Currently, I am only aware of the following issues:

<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.

- 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.
- When loading the package with the `babel` option, you must load the `babel` package before `microtype`.

*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 $\TeX$  in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your  $\TeX$  system.
- ! pdf $\TeX$  error (font expansion): auto expansion is only possible with scalable fonts.  
Automatic font expansion has been improved in pdf $\TeX$  1.40, in that it now not only works with Type 1 fonts but also with TrueType, OpenType and even non-embedded fonts. The above error message indicates either that you are trying to apply expansion to a bitmap (pk) font, which is still not possible, or that the font isn't found at all, e. g., because of missing map entries.
- Warning: pdf $\LaTeX$ : font ptmr8r cannot be expanded (not an included Type1 font) and the PDF viewer complains about a missing font, e. g., Adobe Reader thusly:  
Could not find a font in the Resources dictionary - using Helvetica instead.  
With pdf $\TeX$  versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your  $\TeX$  system is not set up to embed (or 'download') the base PostScript fonts (e. g., Times, Helvetica, Courier). In most  $\TeX$  distributions, this can be changed in the file `updmap.cfg` by setting `pdf $\TeX$ DownloadBase14` to true.
- Warning: pdf $\LaTeX$  (file ecrm1000+20): Font ecrm1000+20 at 1200 not found  
Furthermore, pdf $\TeX$  versions older than 1.40 require Type 1 fonts for automatic font expansion. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. With older pdf $\TeX$  versions, this is only possible if you manually create expanded instances of the fonts.
- ! Font T1/cmr/m/n/10=ecrm1000 at 10.0pt not loaded: Not enough room left.  
Memory parameter 'font\_mem\_size' too small.
- ! TeX capacity exceeded, sorry [maximum internal font number (font\_max)=2000].  
Memory parameter 'font\_max' too small.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf\_mem\_size)=65536].  
Memory parameter 'pdf\_mem\_size' too small (pdf $\TeX$  versions older than 1.30).  
When applying micro-typographic enhancement to a large document with a lot of fonts, pdf $\TeX$  may be running out of some kind of memory. It can be increased by setting the respective parameter to a larger value. For web2c-based systems, e. g.,  $\TeX$ Live, change the settings in `texmf.cnf`, for MiK $\TeX$ , in the file `miktex.ini` (2.4 or older) resp. `pdf $\LaTeX$ .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 pdfTeX programme in the first place, which introduced the micro-typographic extensions and made them available to the TeX world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004](#) and [Thành 2008](#). I also thank him and the rest of the pdfTeX 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  $\text{\TeX}$  typesetting system’, Diss. Masaryk University Brno 2000, in: *TUGBoat*, vol. 21(2000), no. 4, pp. 317–434. (Online at <http://www.tug.org/TUGboat/Articles/tb21-4/tb69thanh.pdf>)

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

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

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

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

$\text{\LaTeX}$ 3 Project Team,  *$\text{\LaTeX}$ 2 $\epsilon$  font selection*, 27 November 2005. (Available from CTAN at [/macros/latex/doc/fntguide.pdf](#))

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

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

## 13 Short History

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

### 2.3 (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 `minil $\text{\textit{tx}}$`  [section 7]

### 2.2 (2007/07/14)

- Improvements to tracking/letterspacing: retain kerning (with pdf $\text{\TeX}$  1.40.4); automatically adjust protrusion settings
- New key ‘no ligatures’ for `\SetTracking` to disable selected or all ligatures (with pdf $\text{\TeX}$  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 pdfTeX 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 LuaTeX (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 pdfTeX version 1.40: tracking/letterspacing, adjustment of interword spacing (`glue`), and additional kerning (new commands `\SetTracking`, `\SetExtraSpacing`, `\SetExtraKerning`; new options ‘`tracking`’, ‘`spacing`’, ‘`kerning`’) [sections 5.3, 5.4, 5.5]
- New commands `\textls` and `\lststyle` for letterspacing, new option ‘`letterspace`’ [sections 3.4, 7]
- New option ‘`babel`’ for automatic micro-typographic adjustment to the selected language [sections 3.5, 6]
- New font sets: ‘`smallcaps`’, ‘`footnotesize`’, ‘`scriptsize`’ [section 4; table 2]
- New package ‘`letterspace`’ providing the commands for robust and hyphenatable letterspacing [section 7]

## 1.9e (2006/07/28)

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

## 1.9d (2006/05/05)

- Support for the Central European QX encoding (inheritance, generic protrusion settings, contributed by Maciej Eder; protrusion settings for Times)
- Protrusion settings for various Euro symbol fonts (Adobe, ITC, marvosym)
- Support for Unicode input in the configuration (`inputenc/utf8`)

## 1.9c (2006/02/02)

- Protrusion settings for URW Garamond

## 1.9a (2005/12/05)

- Defer setup until the end of the preamble; consequently, no need to change font defaults before loading microtype, or to put it the other way round, microtype may now be loaded at any time
- Inside the preamble, `\microtypesetup` accepts all package options [section 3.6]
- Protrusion settings for T5 encoded Charter

## 1.9 (2005/10/28)

- New command `\DisableLigatures` to disable ligatures of fonts (requires pdfTeX version 1.30 or later) [section 8]

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

#### 1.8 (2005/06/23)

- New command `\DeclareMicrotypeSetDefault` to declare the default font sets [section 4]
- New option ‘config’ to load a different configuration file [section 3.5]
- New option ‘unit’ to measure protrusion factors relative to a dimension instead of the character width [section 5.1]
- Renamed commands from `\..MicroType..` to `\..Microtype..`
- Protrusion settings for AMS math fonts
- Protrusion settings for Times in LY1 encoding completed
- The ‘allmath’ font set also includes U encoding
- When using the `ledmac` package, character protrusion will work for the first time ever (requires pdf $\TeX$  version 1.30 or later)

#### 1.7 (2005/03/23)

- Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings [sections 4, 5]
- New command `\LoadMicrotypeFile` to load a font configuration file manually [section 5.7]
- Hook `\Microtype@Hook` for font package authors [section 14.4.3]
- New option ‘verbose=errors’ to turn all warnings into errors
- Warning when running in draft mode

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

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

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

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

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

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

## 1.2 (2004/10/03)

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

## 1.1 (2004/09/21)

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

## 1.0 (2004/09/11)

- First CTAN release



## 14 Implementation

The docstrip modules in this file are:

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

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

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

lua: Code for LuaTeX (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}
```

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

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

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

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

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

```

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

```

These commands also have a starred version.

```

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

```

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

```

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

```

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

```

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

```

`\MT@warning`      Communicate.

```

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

```

### 14.1.1 Debugging

`\tracingmicrotype`      Cases for `\tracingmicrotype`:

```

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

```

```

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

```

\tracingmicrotypeinpdf

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

Cases for \tracingmicrotypeinpdf:

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

```
97 \newcount\tracingmicrotypeinpdf
```

Let's see how it works . . .

```
\tracingmicrotypeinpdf=2
```

\MT@pdf@annot

During font setup, we save the text for the popup in \MT@pdf@annot. (This requires pdf<sub>T</sub><sub>E</sub>X ≥ 1.30.)

\MT@addto@annot

\ifMT@inannot

```

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

```

\iftracingmicrotypeinpdfall

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

```
103 \newif\iftracingmicrotypeinpdfall
```

\MT@show@pdfannot

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

```

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

```

122 `</package>`

### 14.1.2 Requirements

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

0:  $\text{\LaTeX}$

1: `eplain`

2: `miniltx`

```

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

```

`\MT@pdftex@no`  $\text{pdf\TeX}$ '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  $\text{pdf\TeX}$  we're using, if any. `\MT@pdftex@no` will be used throughout the package to respectively do the right thing.

Currently, we have to distinguish seven cases for  $\text{pdf\TeX}$ :

0: not running  $\text{pdf\TeX}$

1:  $\text{pdf\TeX}$  (< 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  $\text{e-TeX}$  ( $\geq$  1.40)

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

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

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

A hack circumventing the  $\text{\TeX}$ Live 2004 hack which undefines the pdf $\text{\TeX}$  primitives in the format in order to hide the fact that pdf $\text{\TeX}$  is being run from the user. This has been *fixed* in  $\text{\TeX}$ Live 2005.

```

144 \ifx\normalpdfptextversion\@undefined \else
145   \let\pdfptextversion\normalpdfptextversion
146   \let\pdfptextrevision\normalpdfptextrevision
147   \let\pdfoutput\normalpdfoutput
148 \fi

```

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

```

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

```

`\MT@clear@options`      If we are not using pdf $\text{\TeX}$  or in case it is too old, we disable everything and exit.

```

187 \def\MT@clear@options{%
188   <plain> \ifnum\MT@plain<2
189     \AtEndOfPackage{\let\@unprocessedoptions\relax}%
190     \let\CurrentOption\empty
191   <plain> \fi
192 }

193 \ifnum\MT@pdfptext@no <
194   <package> 2
195   <letterspace> 6
196   \MT@warning{n}{You
197     \ifcase\MT@pdfptext@no

```

```

198     don't seem to be using pdftex.\MessageBreak
199     `~\MT@MT' only works with pdftex.\MessageBreak
200     Try running `pdflatex' instead of `latex'%
201     \else
202     are using a pdftex version older than
203     <package>      0.14f%
204     <letterspace>  1.40%
205     .\MessageBreak
206     `~\MT@MT' does not work with this version.\MessageBreak
207     Please install a newer version of pdftex%
208     \fi
209   }
210   \MT@clear@options\MT@restore@catcodes
211 \endinput\fi

```

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

```

212 \RequirePackage{keyval}[1997/11/10]
213 <*package>

```

```

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

```

### 14.1.3 Declarations

```

\ifMT@protrusion  These are the global switches ...
\ifMT@expansion  216 \newif\ifMT@protrusion
\ifMT@auto       217 \newif\ifMT@expansion
\ifMT@selected   218 \newif\ifMT@auto
\ifMT@noligatures 219 \newif\ifMT@selected
\ifMT@draft      220 \newif\ifMT@noligatures
\ifMT@spacing    221 \newif\ifMT@draft
\ifMT@kerning    222 \newif\ifMT@spacing
\ifMT@tracking   223 \newif\ifMT@kerning
\ifMT@tracking   224 \newif\ifMT@tracking
\ifMT@tracking   225 \newif\ifMT@babel
\MT@pr@babel     ... and numbers.
\MT@ex@level     226 \let\MT@pr@level\tw@
\MT@pr@factor    227 \let\MT@ex@level\tw@
\MT@ex@factor    228 \let\MT@pr@factor\@m
\MT@sp@factor    229 \let\MT@ex@factor\@m
\MT@kn@factor    230 \let\MT@sp@factor\@m
\MT@kn@factor    231 \let\MT@kn@factor\@m
\MT@pr@unit      Default unit for protrusion settings is character width, for spacing space, for kerning
\MT@sp@unit      (and tracking) 1em.
\MT@kn@unit      232 \let\MT@pr@unit\@empty
\MT@stretch      233 \let\MT@sp@unit\@m@ne
\MT@shrink       234 \def\MT@kn@unit{1em}
\MT@shrink
\MT@step         Expansion settings.
\MT@step         235 \let\MT@stretch\@m@ne
\MT@step         236 \let\MT@shrink \@m@ne
\MT@step         237 \let\MT@step \@m@ne

\MT@pr@min      Minimum and maximum values allowed by pdfTEX.
\MT@pr@max
\MT@ex@min
\MT@ex@max
\MT@sp@min
\MT@sp@max
\MT@kn@min
\MT@kn@max
\MT@tr@min
\MT@tr@max

```

```

238 \def\MT@pr@min{-\@m}
239 \let\MT@pr@max\@m
240 \let\MT@ex@min\z@
241 \let\MT@ex@max\@m
242 \def\MT@sp@min{-\@m}
243 \let\MT@sp@max\@m
244 \def\MT@kn@min{-\@m}
245 \let\MT@kn@max\@m
246 /package
247 \def\MT@tr@min{-\@m}
248 \let\MT@tr@max\@m
249 *package

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

\MT@stretch@default    Default values for expansion.
\MT@shrink@default 251 \def\MT@stretch@default{20 }
\MT@step@default 252 \def\MT@shrink@default{20 }
253 \def\MT@step@default{4 }

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

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

```

#### 14.1.4 Auxiliary Macros

```

\MT@maybe@etex    For definitions that depend on e-TeX features.
259 \ifcase 0%
260   \ifx\TeXversion\@undefined 1\else
261     \ifx\TeXversion\relax 1\else
262       \ifcase\TeXversion 1\fi
263     \fi
264   \fi
265 \else
266   \catcode\^^Q=9 \catcode\^^X=14
267 \fi
268 debug\MT@info{n1}{0}{this is
269 debug\^^Q not
270 debug etex}

\MT@requires@pdftex    For definitions that depend on a particular pdfTeX version.
271 \def\MT@requires@pdftex#1{%
272   \ifnum\MT@pdftex@no<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
273 }
274 debug\MT@requires@pdftex6{\pdftracingfonts=1}\relax

\MT@requires@luatex    For definitions that depend on LuaTeX.
275 *lua
276 \def\MT@requires@luatex{%
277   \ifx\directlua\@undefined\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
278 }
279 debug\MT@info{n10}{this is \MT@requires@luatex}{\not }luatex}

\MT@lua    Communicate with lua.

```



```

280 \def\MT@lua{\directlua{
281   \lua
282 }/package}

```

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

```

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

```

`\MT@glet`     The forgotten primitive.

```

298 \def\MT@glet{\global\let}
299 <*/package>

```

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

```

\MT@gdef@n 300 \def\MT@def@n#1{\expandafter\def\csname #1\endcsname}
301 \def\MT@gdef@n#1{\expandafter\gdef\csname #1\endcsname}

```

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

```

\MT@xdef@n 302 </package>
303 \def\MT@edef@n#1{\expandafter\edef\csname #1\endcsname}
304 <*/package>
305 \def\MT@xdef@n#1{\expandafter\xdef\csname #1\endcsname}

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

312 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
313 <*/package>

```

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

```

314 \def\MT@exp@two@n#1#2#3{%
315   \expandafter\expandafter\expandafter
316   #1\expandafter\expandafter\expandafter
317   {\expandafter#2\expandafter}\expandafter{#3}}

```

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

`\MT@ifdefined@c@T` Wrapper for testing whether command resp. `\csname` sequence is defined. If we  
`\MT@ifdefined@c@TF` are running e-TeX, we will use its primitives `\ifdefined` and `\ifcsname`, which  
`\MT@ifdefined@n@T` decreases memory use substantially.

```

\MT@ifdefined@n@TF 318 \def\MT@ifdefined@c@T#1{%
319 ^^X \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
320 ^^Q \ifx#1\undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
321 }
322 </package>
323 \def\MT@ifdefined@c@TF#1{%
324 ^^X \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
325 <package>^^Q \ifx#1\undefined
326 <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
327 }
328 \def\MT@ifdefined@n@T#1{%
329 ^^X \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
330 <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
331 <package>^^Q \expandafter\@gobble\else\expandafter\@firstofone\fi
332 }
333 <*package>
334 \def\MT@ifdefined@n@TF#1{%
335 ^^X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
336 ^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
337 ^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
338 }

```

`\MT@detokenize@n` Translate a macro into a token list. With e-TeX, we can use `\detokenize` (and  
`\MT@detokenize@c` `\expandafter\string` to get rid of the trailing space). The  $\not\in$ -TeX version requires  
some more fiddling (and the `\string` isn't perfect, of course).

```

339 \def\MT@detokenize@n#1{%
340 ^^X \detokenize\expandafter{%
341 \string#1%
342 ^^X }%
343 }
344 \def\MT@detokenize@c#1{%
345 ^^X \detokenize\expandafter\expandafter\expandafter{\expandafter\string#1}%
346 ^^Q \MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty
347 }

```

`\MT@ifempty` Test whether argument is empty.

```

348 </package>
349 \begingroup
350 \catcode`\%=12
351 \catcode`\&=14
352 \gdef\MT@ifempty#1{%
353 \if %#1%&
354 \expandafter\@firstoftwo
355 \else
356 \expandafter\@secondoftwo
357 \fi
358 }
359 \endgroup
360 <*package>

```

`\MT@ifint` Test whether argument is an integer, using an old trick by Mr. Arseneau, or the  
latest and greatest from pdfTeX or LuaTeX (which also allows negative numbers, as  
required by the `letterspace` option).

```

361 \MT@requires@pdftex6{
362 <*lua>
363 \MT@requires@luatex{

```

```

364 \def\MT@ifint#1{%
365   \csname
366     \MT@lua{if string.find("\luaescapestring{#1}", "^-[0-9]+ *$")
367       then tex.write("@firstoftwo")
368       else tex.write("@secondoftwo")
369     end}%
370   \endcsname
371 }
372 }{
373 \lua
374 \package
375 \def\MT@ifint#1{%
376   \ifcase\pdfmatch{^-[0-9]+ *$}{#1}\relax
377   \expandafter\@secondoftwo
378   \else
379   \expandafter\@firstoftwo
380   \fi
381 }
382 \package
383 \lua }
384 }{
385 \def\MT@ifint#1{%
386   \if!\ifnum9<1#1!\else?\fi
387   \expandafter\@firstoftwo
388   \else
389   \expandafter\@secondoftwo
390   \fi
391 }
392 }

```

`\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.)

```

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

```

```

423     \aftergroup\@firstoftwo
424     \fi
425 }%
426 }
427 }

\MT@ifdim    Test floating point numbers.
428 \def\MT@ifdim#1#2#3{%
429   \ifdim #1\p@ #2 #3\p@
430     \expandafter\@firstoftwo
431   \else
432     \expandafter\@secondoftwo
433   \fi
434 }

\MT@ifstreq  Test whether two strings (fully expanded) are equal.
435 \MT@requires@pdftex5{
436   *lua
437   \MT@requires@lualatex{
438     \def\MT@ifstreq#1#2{%
439       \csname
440         \MT@lua{if "\luaescapestring{#1}" == "\luaescapestring{#2}"
441           then tex.write("@firstoftwo")
442           else tex.write("@secondoftwo")
443         end}%
444       \endcsname
445     }
446   }{
447     /lua
448     \def\MT@ifstreq#1#2{%
449       \ifcase\pdfstrcmp{#1}{#2}\relax
450       \expandafter\@firstoftwo
451     \else
452       \expandafter\@secondoftwo
453     \fi
454   }
455   lua }
456 }{
457   \def\MT@ifstreq#1#2{%
458     \edef\MT@res@a{#1}%
459     \edef\MT@res@b{#2}%
460     \ifx\MT@res@a\MT@res@b
461       \expandafter\@firstoftwo
462     \else
463       \expandafter\@secondoftwo
464     \fi
465   }
466 }

\MT@xadd    Add item to a list.
467 \def\MT@xadd#1#2{%
468   \ifx#1\relax
469     \xdef#1{#2}%
470   \else
471     \xdef#1{#1#2}%
472   \fi
473 }

\MT@xaddb   Add item to the beginning.
474 \def\MT@xaddb#1#2{%
475   \ifx#1\relax
476     \xdef#1{#2}%

```

```

477 \else
478 \xdef#1{#2#1}%
479 \fi
480 }

\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@ 481 </package>
\MT@clist@function 482 \def\MT@map@clist@n#1#2{%
483 \ifx\@empty#1\else
\MT@clist@break 484 \def\MT@clist@function##1{#2}%
485 \MT@map@clist@#1,\@nil,\@nnil
486 \fi
487 }
488 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
489 \def\MT@map@clist@#1,%
490 \ifx\@nil#1%
491 \expandafter\MT@clist@break
492 \fi
493 \MT@clist@function{#1}%
494 \MT@map@clist@
495 }
496 \let\MT@clist@function\@gobble
497 \def\MT@clist@break#1\@nnil{}
498 <*/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@ 499 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break 500 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
501 \def\MT@map@tlist@#1#2{%
502 \ifx\@nnil#2\else
503 #1{#2}%
504 \expandafter\MT@map@tlist@
505 \expandafter#1%
506 \fi
507 }
508 \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 509 \newif\ifMT@inlist@
510 \def\MT@in@clist#1#2{%
511 \def\MT@res@a#1,#1,##2##3\@nnil{%
512 \ifx##2\@empty
513 \MT@inlist@false
514 \else
515 \MT@inlist@true
516 \fi
517 }%
518 \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
519 }

\MT@rem@from@clist Remove item <#1> from comma list <#2>. This is basically \@removeelement from
ltcncrl.dtx. Using \pdfmatch and \pdflastmatch here would be really slow!
520 \def\MT@rem@from@clist#1#2{%
521 \def\MT@res@a#1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
522 \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
523 \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
524 }

\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.

```

525 \def\MT@in@tlist#1#2{%
526   \MT@inlist@false
527   \def\MT@res@a{#1}%
528   \MT@map@tlist@c#2\MT@in@tlist@
529 }
530 \def\MT@in@tlist@#1{%
531   \edef\MT@res@b{#1}%
532   \ifx\MT@res@a\MT@res@b
533     \MT@inlist@true
534     \expandafter\MT@tlist@break
535   \fi
536 }

\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@@ 537 \def\MT@in@rlist#1{%
\MT@size@name 538   \MT@inlist@false
539   \MT@map@tlist@c#1\MT@in@rlist@
540 }
541 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
542 \def\MT@in@rlist@@#1#2#3{%
543   \MT@ifdim{#2}=\m@ne{%
544     \MT@ifdim{#1}=\MT@size
545     \MT@inlist@true
546     \relax
547   }%
548   \MT@ifdim\MT@size<{#1}\relax{%
549     \MT@ifdim\MT@size<{#2}%
550     \MT@inlist@true
551     \relax
552   }%
553 }%
554 \ifMT@inlist@
555   \def\MT@size@name{#3}%
556   \expandafter\MT@tlist@break
557 \fi
558 }

\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 559 </package>
560 \def\MT@loop#1\MT@repeat{%
561   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
562   \MT@iterate \let\MT@iterate\relax
563 }
564 \let\MT@repeat\fi

\MT@while@num    Execute <#3> from <#1> up to (excluding) <#2> (much faster than LATEX's \whilenum).
565 \def\MT@while@num#1#2#3{%
566   \@tempcnta#1\relax
567   \MT@loop #3%
568   \advance\@tempcnta \@ne
569   \ifnum\@tempcnta < #2\MT@repeat
570 }

\MT@do@font    Execute <#1> 256 times.
571 \def\MT@do@font{\MT@while@num\z@\@ccclvi}
572 <*package>

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

```

slightly faster.

```
573 \newcount\MT@count
574 \def\MT@increment#1{%
575   ^^X \edef#1{\number\numexpr #1 + 1\relax}%
576   ^^Q \MT@count=#1\relax
577   ^^Q \advance\MT@count \@ne
578   ^^Q \edef#1{\number\MT@count}%
579 }
```

`\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.

```
580 \def\MT@scale#1#2#3{%
581   ^^Q \multiply #1 #2\relax
582   \ifnum #3 = \z@
583     ^^X #1=\numexpr #1 * #2\relax
584   \else
585     ^^X #1=\numexpr #1 * #2 / #3\relax
586     ^^Q \divide #1 #3\relax
587   \fi
588 }
```

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

```
\MT@abbr@ex
\MT@abbr@pr@c 589 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 590 \def\MT@abbr@ex{expansion}
\MT@abbr@pr@inh 591 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@ex@inh 592 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh 593 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@nl 594 \def\MT@abbr@ex@inh{expansion inheritance}
\MT@abbr@sp 595 \def\MT@abbr@nl{noligatures}
\MT@abbr@sp@c 596 \def\MT@abbr@sp{spacing}
\MT@abbr@sp@inh 597 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@sp@inh 598 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@kn 599 \def\MT@abbr@kn{kerning}
\MT@abbr@kn@c 600 \def\MT@abbr@kn@c{kerning codes}
\MT@abbr@kn@c 601 \def\MT@abbr@kn@inh{kerning inheritance}
\MT@abbr@kn@inh 602 \def\MT@abbr@tr{tracking}
\MT@abbr@tr@c 603 \def\MT@abbr@tr@c{tracking amount}
```

These we also need the other way round.

```
\MT@rbba@protrusion
\MT@rbba@expansion 604 \def\MT@rbba@protrusion{pr}
\MT@rbba@spacing 605 \def\MT@rbba@expansion{ex}
\MT@rbba@kerning 606 \def\MT@rbba@spacing{sp}
\MT@rbba@tracking 607 \def\MT@rbba@kerning{kn}
\MT@rbba@tracking 608 \def\MT@rbba@tracking{tr}
```

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

```
\MT@features@long 609 \def\MT@features{pr,ex,sp,kn,tr}
610 \def\MT@features@long{protrusion,expansion,spacing,kerning,tracking}
```

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

```
611 \def\MT@is@feature#1{%
612   \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
613   \ifMT@inlist@
614     \expandafter\@firstofone
```

```

615 \else
616   \MT@error{\@tempa' is not an available micro-typographic\MessageBreak
617     feature. Ignoring #1}{Available features are: \MT@features@long'.}%
618   \expandafter\gobble
619 \fi
620 }

```

### 14.1.5 Compatibility

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

- `\pickup@font`
- `\do@subst@correction`
- `\add@accent` (all in section 14.2.9)
- `\showhyphens` (in section 14.4.5)

The wordcount package redefines the font-switching commands, which will break microtype. Since microtype doesn't have an effect on the number of words in the document anyway, we will simply disable ourselves.

```

621 \ifl@aded{tex}{wordcount}{%
622   \MT@warning@nl{Detected the 'wordcount' utility.\MessageBreak
623     Disabling \MT@MT', since it wouldn't work}%
624   \MT@clear@options\MT@restore@catcodes\endinput}\relax

```

`\MT@setup@` The setup is deferred until the end of the preamble. This has a couple of advantages: `\microtypesetup` can be used to change options later on in the preamble, and fonts don't have to be set up before microtype.

```

625 \<package>
626 \<plain>\ifnum\MT@plain<2
627 \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.

```

628 \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).

```

629 \AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@ \@empty}

```

Don't hesitate with `miniltx`.

```

630 \<plain>\else \let\MT@addto@setup \@firstofone \fi

```

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

```

631 \def\MT@with@package@T#1{\ifpackage@loaded{#1}\@firstofone\@gobble}
632 \<package>

```

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

```

633 \def\MT@with@babel@and@T#1{%
634   \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
635     \@expandtwoargs\MT@in@clist{#1}
636     {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
637     \ifMT@in@list@ \expandafter\@gobble\fi
638   }\@gobble
639 }

```

Don't load `letterspace`.

```

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

```



\MT@ledmac@setup  
 \MT@led@unhbox@line  
 \MT@led@kern

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).

```

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

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

```

674 \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`.)

```

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

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

in `\percentsign`.

```

676 \MT@if@false
677 \MT@with@babel@and@T{spanish} \MT@if@true
678 \MT@with@babel@and@T{galician} \MT@if@true
679 \MT@with@babel@and@T{mexican} \MT@if@true
680 \ifMT@if@MT@ifdefined@c@T\percentsign{\let%\percentsign}\fi

```

Using `\@disablequotes`, we can restore the original meaning of all characters made active by `csquotes`. (It would be doable for older versions, too, but we won't bother.)

```

681 \MT@with@package@T{csquotes}{%
682 \ifpackage@later{csquotes}{2005/05/11}\@disablequotes\relax}%

```

`hyperref` redefines `\%` and `\#` inside a `\url`. We restore the original meanings (which we can only hope are correct). Same for `tex4ht`.

```

683 \MT@if@false
684 \MT@with@package@T{hyperref} \MT@if@true
685 \MT@with@package@T{tex4ht} \MT@if@true
686 \ifMT@if@MT@restore@p@h\fi
687 }

```

Check again at the end of the preamble.

```

688 </package>
689 \MT@addto@setup{%
690 <*package>

```

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

```

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

```

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

```

699 \let\MT@setupfont@hook\empty
700 \MT@if@false
701 \MT@with@babel@and@T{spanish} \MT@if@true
702 \MT@with@babel@and@T{galician} \MT@if@true
703 \MT@with@babel@and@T{mexican} \MT@if@true
704 \ifMT@if@
705 \g@addto@macro\MT@setupfont@hook{%
706 \MT@ifdefined@c@T\percentsign{\let%\percentsign}}%
707 \fi
708 \MT@with@package@T{csquotes}{%
709 \ifpackage@later{csquotes}{2005/05/11}{%
710 \g@addto@macro\MT@setupfont@hook\@disablequotes
711 }{%
712 \MT@warning@n1{%
713 Should you receive warnings about unknown slot\MessageBreak
714 numbers, try upgrading the `csquotes' package}%
715 }%
716 }

```

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

```

717 \MT@if@false
718 </package>

```

```

719 <plain> \ifnum\MT@plain=0
720 \MT@with@package@T{hyperref}{%
721 \pdfstringdefDisableCommands{%
722 <*package>
723 \let\pickup@font\MT@orig@pickupfont
724 \let\textmicrotypecontext\@secondoftwo
725 \let\microtypecontext\@gobble
726 </package>
727 \def\lsstyle{\pdfstringdefWarn\lsstyle}%
728 \def\textls#1#{\pdfstringdefWarn\textls}%
729 }%
730 <package> \MT@if@true
731 }
732 <plain> \fi
733 <*package>
734 \MT@with@package@T{tex4ht}\MT@if@true
735 \ifMT@if@lg@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi

```

The listings package makes numbers and letters active,

```

736 \MT@with@package@T{listings}{%
737 \g@addto@macro\MT@cfg@catcodes{%
738 \MT@while@num{"30"}{"3A"}{\catcode\@tempcnta 12\relax}%
739 \MT@while@num{"41"}{"5B"}{\catcode\@tempcnta 11\relax}%
740 \MT@while@num{"61"}{"7B"}{\catcode\@tempcnta 11\relax}%
741 }%

```

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

```

742 \g@addto@macro\MT@setupfont@hook{%
743 \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.

```

744 \let\lst@ProcessLetter\@empty
745 }%
746 }

```

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.

```

747 </package>
748 <plain> \ifnum\MT@plain=0
749 \MT@with@package@T{soul}{%
750 \soulregister\lsstyle 0%
751 \soulregister\textls 1%
752 }

```

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

```

753 <*plain>
754 \else
755 \ifx\Soul@\undefined\else
756 \soulregister\lsstyle 0%
757 \soulregister\textls 1%
758 \fi
759 \fi
760 </plain>
761 <*package>

```

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

had only one argument.

```

762 \MT@with@package@T{pinyin}{%
763   \let\MT@orig@py@macron\py@macron
764   \ifpackage@later{pinyin}{2006/10/17}{% 4.7.0
765     \def\py@macron#1#2{%
766       \let\pickup@font\MT@orig@pickupfont
767       \MT@orig@py@macron{#1}{#2}%
768       \let\pickup@font\MT@pickupfont}%
769     }{%
770       \def\py@macron#1{%
771         \let\pickup@font\MT@orig@pickupfont
772         \MT@orig@py@macron{#1}%
773         \let\pickup@font\MT@pickupfont}%
774       }%
775     }
776 \end{package}
777 }
778 \end{package}

```

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

```

779 \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`).

```

780 \def\MT@setupfont{%

```

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

```

781 \MT@setupfont@hook
782 }

```

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

```

783 \MT@requires@pdftex7
784 {\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!

```

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

```

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

```

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

```

790 % \ifx\f@encoding\cf@encoding\else\@enc@update\fi
791 }

```

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

```

792 \MT@requires@pdftex6
793 {\g@addto@macro\MT@setupfont\MT@tracking}\relax
794 \g@addto@macro\MT@setupfont{%
795   \MT@check@font

```

```

796 \ifMT@inlist@
797 <debug>\MT@show@pdfannot2%
798 \else
799 \MT@info{Setting up font `\'MT@@font'\on@line}%

```

Now we can begin setting up the font for all features that the current pdfTeX provides. The following commands are \let to \relax if the respective feature is disabled via package options.

For versions older than 1.20, protrusion has to be set up first, beginning with 1.20, the order doesn't matter.

```

800 \MT@protrusion
801 \MT@expansion
802 }

```

Interword spacing and kerning (pdfTeX 1.40).

```

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

```

Disable ligatures (pdfTeX 1.30).

```

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

```

Debugging.

```

808 <debug>\MT@show@pdfannot1%

```

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

```

809 \MT@register@font
810 \fi
811 }

```

\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.

```

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

```

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

```

815 \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.

```

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

```

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

```

825 \MT@map@clist@c\MT@active@features{%
826 \expandafter\ifx\csname MT@%nameuse{MT@abbr@##1}\endcsname\relax\else
827 \def@tempa{##1}%
828 \expandafter\MT@map@tlist@c
829 \csname MT@##1@doc@contexts\endcsname
830 \MT@rem@from@list

```

```

831     \fi
832   }%
833   \fi
834   \MT@exp@two@c\let\MT@font\MT@font@copy

```

We only need the font identifier for letterspacing.

```

835   \let\font@name\MT@font@copy

```

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

```

836   \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
837 }

```

\MT@rem@from@list

```

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

```

*Here's the promised dirty trick* for users of older pdfTeX versions, which works around the problem that the use of the same font with different expansion parameters is prohibited. If you do not want to create a clone of the font setup (this would require duplicating the tfm/vf files under a new name, and writing new fd files and map entries), you can load a minimally larger font for the paragraph in question. E. g., for a document typeset in 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 845 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%

```

```

846   \def\MT@encoding{#1}%

```

```

\MT@series 847   \def\MT@family {#2}%

```

```

\MT@shape 848   \def\MT@series {#3}%

```

```

\MT@size 849   \def\MT@shape {#4}%

```

```

850   \def\MT@size {#5}%

```

\MT@familyalias      Alias family?

```

851   \MT@ifdefined@n@TF{MT@\MT@family @alias}%

```

```

852   {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%

```

```

853   {\let\MT@familyalias\@empty}%

```

```

854 }

```

```

\ifMT@do    We check all features of the current font against the lists of the currently active
\MT@feat    font set, and set \ifMT@do accordingly.
\MT@maybe@do 855 \newif\ifMT@do
               856 \def\MT@maybe@do#1{%
               (but only if the feature isn't globally set to false)
               857 \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).
               858 \MT@dotrue
               859 \edef\@tempa{\csname MT@#1\setname\endcsname}%
               860 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
               861 \MT@ifdefined@n@TF{MT@checklist@#1}%
               862 {\csname MT@checklist@#1\endcsname}%
               863 {\MT@checklist@{#1}}%
               864 {#1}%
               865 }%
               866 \else
               867 \MT@dofalse
               868 \fi
               869 \ifMT@do
               \MT@feat stores the current feature.
               870 \def\MT@feat{#1}%
               871 \csname MT@set@#1\codes\endcsname
               872 \else
               873 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
               874 \fi
               875 }
\MT@dinfo@list
               876 <debug>\def\MT@dinfo@list#1#2#3{\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#1}:
               877 <debug> #2 \ifx\\#3\\list empty\else \@nameuse{MT@#2}' #3 list\fi}%
               878 <debug>}}
\MT@checklist@ The generic test (<#1> is the axis, <#2> the feature, \@tempa contains the set name).
               879 \def\MT@checklist@#1#2{%
               880 <!debug> \MT@ifdefined@n@T
               881 <debug> \MT@ifdefined@n@TF
               882 {MT@#2list@#1@\@tempa}{%
               Begin an \expandafter orgy to test whether the font attribute is in the list.
               883 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
               884 \csname MT@#1\expandafter\endcsname
               885 \csname MT@#2list@#1@\@tempa\endcsname
               886 \ifMT@inlist@
               887 <debug>\MT@dinfo@list{#2}{#1}{in}%
               888 \MT@dotrue
               889 \else
               890 <debug>\MT@dinfo@list{#2}{#1}{not in}%
               891 \MT@dofalse
               892 \expandafter\MT@clist@break
               893 \fi
               894 }%
               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.
               895 <debug> {\MT@dinfo@list{#2}{#1}{}}%
               896 }

```

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

```

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

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

```

925 \def\MT@checklist@size#1{%
926 <!--debug--> \MT@ifdefined@n@T
927 <!--debug--> \MT@ifdefined@n@TF
928     {MT@#1list@size@ \@tempa}{%
929     \expandafter\MT@in@rlist\csname MT@#1list@size@ \@tempa\endcsname
930     \ifMT@inlist@
931 <!--debug-->\MT@info@list{#1}{size}{in}%
932     \MT@dotrue
933     \else
934 <!--debug-->\MT@info@list{#1}{size}{not in}%
935     \MT@dofalse
936     \expandafter\MT@clist@break
937     \fi
938 }%
939 <!--debug--> {\MT@info@list{#1}{size}}}%
940 }
```

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

```

941 \def\MT@checklist@font#1{%
942 <!--debug--> \MT@ifdefined@n@T
943 <!--debug--> \MT@ifdefined@n@TF
944     {MT@#1list@font@ \@tempa}{%
```

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

```

945     \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
946     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
947     \@tempb \csname MT@#1list@font@ \@tempa\endcsname
948     \ifMT@inlist@
949 <!--debug-->\MT@info@list{#1}{font}{in}%
950     \expandafter\MT@clist@break
```



```

951     \else
952 <debug> \MT@info@list{#1}{font}{not in}%
953     \MT@dofalse
954     \fi
955 }%
956 <debug> { \MT@info@list{#1}{font}{} }%
957 }

```

### 14.2.1 Protrusion

```

\MT@protrusion    Set up for protrusion?
958 \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.
959 \def\MT@set@pr@codes{%
    Check whether and if, which list should be applied to the current font.
960     \MT@if@list@exists{%
961         \MT@get@font@dimen@six
962         \MT@get@opt
963         \MT@reset@pr@codes

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

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

    Load additional lists?
966     \MT@load@list\MT@pr@c@name
967     \MT@set@listname

    Load the main list.
968     \MT@let@cn\@tempc{\MT@pr@c@\MT@pr@c@name}%
969     \expandafter\MT@set@codes\@tempc,\relax,%
970 } \MT@reset@pr@codes
971 }

\MT@gobble@settings
972 \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
\MT@dimen@six            settings (for example, the dsfont fonts don't specify this dimension; this is probably
                        a bug).
973 \def\MT@get@font@dimen@six{%
974     \ifnum\fontdimen6\MT@font=\z@
975         \MT@warning@nl{%
976             Font ``\MT@font' does not specify its\MessageBreak
977             \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
978             \@nameuse{\MT@abbr@\MT@feat} will not work with this font}%
979         \expandafter\MT@gobble@settings
980     \else
981         \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
982     \fi
983 }

\MT@set@all@pr    Set all protrusion codes of the font.
984 \def\MT@set@all@pr#1#2{%
985 <debug> \MT@info@nl{3}{-- lp/rp: setting all to #1/#2}%

```

```

986 \let\MT@temp\empty
987 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lcode\MT@font\@tempcnta=#1\relax}}%
988 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rcode\MT@font\@tempcnta=#2\relax}}%
989 \MT@do@font\MT@temp
990 }

\MT@reset@pr@codes@    All protrusion codes are zero for new fonts. However, if we have to reload the font
\MT@reset@pr@codes    due to different contexts, we have to reset them. This command will be changed by
                        \microtypecontext if necessary.
991 \def\MT@reset@pr@codes@{\MT@set@all@pr@z@z@}
992 \let\MT@reset@pr@codes\relax

\MT@the@pr@code        If the font is letterspaced, we have to add half the letterspacing amount to the
\MT@the@pr@code@tr     margin kerns. This will be activated in \MT@set@tr@codes.
993 \def\MT@the@pr@code{\@tempcntb}
994 \MT@requires@pdftex6{
995   \def\MT@the@pr@code@tr{%
996     \numexpr\@tempcntb+\MT@letterspace@/2\relax
997   }
998 }\relax

\MT@set@codes          Split up the values and set the codes.
999 \def\MT@set@codes#1,{%
1000   \ifx\relax#1\empty\else
1001     \MT@split@codes #1==\relax
1002     \expandafter\MT@set@codes
1003   \fi
1004 }

\MT@split@codes        The keyval package would remove spaces here, which we needn't do since
                        \SetProtrusion ignores spaces in the protrusion list anyway. \MT@get@char@unit
                        may mean different things.
1005 \def\MT@split@codes#1=#2=#3\relax{%
1006   \def\@tempa{#1}%
1007   \ifx\@tempa\empty \else
1008     \MT@get@slot
1009     \ifnum\MT@char > \m@ne
1010       \MT@get@char@unit
1011       \csname MT@\MT@feat @split@val\endcsname#2\relax
1012     \fi
1013   \fi
1014 }

\MT@pr@split@val
1015 \def\MT@pr@split@val#1,#2\relax{%
1016   \def\@tempb{#1}%
1017   \MT@ifempty\@tempb\relax{%
1018     \MT@scale@to@em
1019     \lcode\MT@font\MT@char=\MT@the@pr@code
1020     <debug>\MT@info@n1{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char: [#1]}%
1021   }%
1022   \def\@tempb{#2}%
1023   \MT@ifempty\@tempb\relax{%
1024     \MT@scale@to@em
1025     \rcode\MT@font\MT@char=\MT@the@pr@code
1026     <debug>\MT@info@n1{4}{;;; rp (\MT@char): \number\rendcode\MT@font\MT@char: [#2]}%
1027   }%

```

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

```

1028 \MT@ifdefined@c@T\MT@pr@inh@name{%
1029 \MT@ifdefined@n@T\MT@inh@MT@pr@inh@name @\MT@char @}{%
1030 \expandafter\MT@map@tlist@c
1031 \csname MT@inh@MT@pr@inh@name @\MT@char @\endcsname
1032 \MT@set@pr@heirs
1033 }%
1034 }%
1035 }

```

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

```

1036 \MT@requires@pdftex3{
1037 \def\MT@scale@to@em{%
1038 \@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`.

```

1039 \MT@scale\@tempcntb \@tempb \MT@dimen@six
1040 \ifnum\@tempcntb=\z@ \else
1041 \MT@scale@factor
1042 \fi
1043 }

```

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

```

1044 \def\MT@get@charwd{%
1045 ^^X \MT@count=\fontcharwd\MT@font\MT@char\relax
1046 ^^Q \setbox\z@=\hbox{\MT@font \char\MT@char}%
1047 ^^Q \MT@count=\wd\z@
1048 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1049 }

```

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`.

```

1050 \MT@requires@pdftex6{
1051 \g@addto@macro\MT@get@charwd{%
1052 \MT@ifdefined@c@T\MT@letterspace@
1053 {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1054 }
1055 }\relax
1056 }}

```

No adjustment with versions 0.14f and 0.14g.

```

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

```

1062 }

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

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

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 \def\MT@info@missing@char{%

1075 \MT@info@n1{Character '\the\MT@toks'

1076 ^^X \iffontchar\MT@font\MT@char

1077 has a width of 0pt

1078 ^^X \else is missing\fi

1079 ^^Q \MessageBreak (it's probably missing)

1080 \MessageBreak in font '\MT@font'. \MessageBreak

1081 Ignoring protrusion settings for this character}%

1082 }

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

1083 \def\MT@scale@factor{%

1084 \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else

1085 \expandafter\MT@scale\expandafter \@tempcntb

1086 \csname MT@\MT@feat @factor@\endcsname \@m

1087 \fi

1088 \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax

1089 \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @max\endcsname

1090 \else

1091 \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax

1092 \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @min\endcsname

1093 \fi

1094 \fi

1095 }

\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.

1096 \def\MT@warn@code@too@large#1{%

1097 \@tempcnta=#1\relax

1098 \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else

1099 \expandafter\MT@scale\expandafter \@tempcnta\expandafter

1100 \@m \csname MT@\MT@feat @factor@\endcsname

1101 \fi

1102 \MT@scale\@tempcnta \MT@dimen@six \MT@count

1103 \MT@warning@n1{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space

1104 is too large for character \MessageBreak

1105 '\the\MT@toks' in \MT@curr@list@name. \MessageBreak

1106 Setting it to the maximum of \number\@tempcnta}%

1107 \@tempcntb=#1\relax

1108 }

\MT@get@opt The optional argument to the configuration commands (except for \SetExpansion,

which is being dealt with in \MT@get@ex@opt).

```
1109 \def\MT@get@opt{%
1110   \MT@set@listname
```

\MT@pr@factor@      Apply a factor?

```
\MT@sp@factor@ 1111 \MT@ifdefined@n@TF{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 1112 \MT@let@nn{MT@MT@feat @factor@}
1113 {MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @factor}%
1114 \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@MT@feat} codes by
1115 \number\csname MT@MT@feat @factor@\endcsname/1000}%
1116 }{%
1117 \MT@let@nn{MT@MT@feat @factor@}{MT@MT@feat @factor}%
1118 }%
```

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

```
\MT@sp@unit@
\MT@kn@unit@ 1119 \MT@ifdefined@n@TF{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @unit}{%
1120 \MT@let@nn{MT@MT@feat @unit@}%
1121 {MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @unit}%
1122 \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\empty
1123 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
1124 relative to character widths}%
1125 \else
1126 \expandafter\ifx\csname MT@MT@feat @unit@\endcsname@m@ne
1127 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
1128 relative to width of space}%
1129 \fi
1130 \fi
1131 }{%
1132 \MT@let@nn{MT@MT@feat @unit@}{MT@MT@feat @unit}%
1133 }%
```

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

```
1134 \let\MT@get@char@unit\relax
1135 \let\MT@get@space@unit@gobble
1136 \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\empty
1137 \let\MT@get@char@unit\MT@get@charwd
1138 \else
1139 \expandafter\ifx\csname MT@MT@feat @unit@\endcsname@m@ne
1140 \let\MT@get@space@unit\MT@get@font@dimen
1141 \else
1142 \expandafter\MT@get@unit\csname MT@MT@feat @unit@\endcsname
1143 \fi
1144 \fi
```

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

```
1145 \MT@ifdefined@n@T{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @preset}{%
1146 \csname MT@preset@MT@feat\endcsname
1147 \MT@let@nc{MT@reset@MT@feat @codes}\relax
1148 }%
1149 }
```

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

```
1150 \def\MT@get@unit#1{%
1151 \expandafter\MT@get@unit@#1 e!\@nil
```

```

1152 \ifx\x\@empty\else\let#1\x\fi
1153 \@defaultunits\@tempdima#1 pt\relax\@nnil
1154 \ifdim\@tempdima=\z@
1155   \MT@warning@n1{%
1156     Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
1157     width. Setting factors of list \@nameuse{MT@\MT@feat @c@name}'\MessageBreak
1158     relative to character widths instead}%
1159   \let#1\@empty
1160   \let\MT@get@char@unit\MT@get@charwd
1161 \else
1162   \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
1163     to \the\@tempdima}%
1164   \MT@count=\@tempdima\relax
1165 \fi
1166 }
1167 \def\MT@get@unit@#1e#2#3\@nil{%
1168 \ifx\#3\\\let\x\@empty \else
1169   \if m#2%
1170     \edef\x{#1\fontdimen6\MT@font}%
1171   \else
1172     \if x#2%
1173       \edef\x{#1\fontdimen5\MT@font}%
1174     \fi
1175   \fi
1176 \fi
1177 }

```

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

```
1178 \def\MT@set@inputenc#1{%
```

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

```

1179   \def\MT@cat{#1}%
1180   \edef\@tempa{MT@\MT@feat @#1@\csname MT@\MT@feat @#1@name\endcsname @inputenc}%
1181   \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1182 }

```

\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.

```

1183 \MT@addto@setup{%
1184   \ifpackageloaded{inputenc}{
1185     \ifpackageafter{inputenc}{2006/02/22}{
1186       \def\MT@set@inputenc@{%
1187         \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1188         \MT@load@inputenc
1189       }
1190     }{
1191       \let\MT@set@inputenc@\MT@load@inputenc
1192     }
1193   }{
1194     \def\MT@set@inputenc@{%
1195       \MT@warning@n1{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1196         \MessageBreak package isn't loaded. Ignoring input encoding}%
1197     }
1198   }
1199 }

```

\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.

```

1200 \def\MT@load@inputenc{%
1201   \MT@cfg@catcodes
1202   (debug)\MT@edinfo@n1{1}{loading input encoding: \@nameuse{\@tempa}}%

```

```

1203 \inputencoding{\@nameuse{\@tempa}}%
1204 }

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

1205 \def\MT@set@pr@heirs#1{%
1206   \lcode\MT@font#1=\lcode\MT@font\MT@char
1207   \rcode\MT@font#1=\rcode\MT@font\MT@char
1208   <debug>\MT@info@nl{2}{-- heir of \MT@char: #1}%
1209   <debug>\MT@info@nl{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
1210   <debug>                                     \number\rcode\MT@font\MT@char}%
1211 }

\MT@preset@pr      Preset characters. Presetting them relative to their widths is not allowed.
\MT@preset@pr@ 1212 \def\MT@preset@pr{%
1213   \expandafter\expandafter\expandafter\MT@preset@pr@
1214   \csname MT@pr@cc@MT@pr@cc@name @preset\endcsname\@nil
1215 }
1216 \def\MT@preset@pr@#1,#2\@nil{%
1217   \ifx\MT@pr@unit@\@empty
1218     \MT@warn@preset@towidth{pr}%
1219     \let\MT@preset@aux\MT@preset@aux@factor
1220   \else
1221     \def\MT@preset@aux{\MT@preset@aux@space2}%
1222   \fi
1223   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1224   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1225   \MT@set@all@pr\@tempa\@tempb
1226 }

\MT@preset@aux    Auxiliary macro for presetting. Store value <#1> in macro <#2>.
\MT@preset@aux@factor 1227 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 1228   \@tempcntb=#1\relax
1229   \MT@scale@factor
1230   \edef#2{\number\@tempcntb}%
1231 }
1232 \def\MT@preset@aux@space#1#2#3{%
1233   \def\@tempb{#2}%
1234   \MT@get@space@unit#1%
1235   \MT@scale@to@em
1236   \edef#3{\number\@tempcntb}%
1237 }

\MT@warn@preset@towidth

1238 \def\MT@warn@preset@towidth#1{%
1239   \MT@warning@nl{%
1240     Cannot preset characters relative to their widths\MessageBreak
1241     for \@nameuse{MT@abbr@#1} list ``\@nameuse{MT@#1@c@name}'. Presetting them%
1242     \MessageBreak relative to lem instead}%
1243 }

```

### 14.2.2 Expansion

```

\MT@expansion      Set up for expansion?
1244 \def\MT@expansion{\MT@maybe@do{ex}}

\MT@set@ex@codes@es    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).
1245 \def\MT@set@ex@codes@es{%

```

```

1246 \MT@if@list@exists{%
1247   \MT@get@ex@opt
1248   \let\MT@get@char@unit\relax
1249   \MT@reset@ef@codes
1250   \MT@get@inh@list
1251   \MT@set@inputenc{c}%
1252   \MT@load@list\MT@ex@cname
1253   \MT@set@listname
1254   \MT@let@cn\@tempc{MT@ex@c@\MT@ex@cname}%
1255   \expandafter\MT@set@codes\@tempc,\relax,%
1256   \MT@expandfont
1257 } \relax
1258 }

```

`\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.

```

1259 \newif\ifMT@nonselected
1260 \def\MT@set@ex@codes@n{%
1261   \MT@nonselectedtrue
1262   \MT@if@list@exists
1263   \MT@get@ex@opt
1264   {%
1265     \let\MT@stretch@ \MT@stretch
1266     \let\MT@shrink@ \MT@shrink
1267     \let\MT@step@ \MT@step
1268     \let\MT@auto@ \MT@auto
1269     \let\MT@ex@factor@ \MT@ex@factor
1270   }%
1271   \MT@reset@ef@codes
1272   \MT@expandfont
1273   \MT@nonselectedfalse
1274 }

```

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

```
1275 \let\MT@set@ex@codes\MT@set@ex@codes@n
```

`\MT@expandfont` Expand the font.

```

1276 \def\MT@expandfont{%
1277   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1278 }

```

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

```

1279 \def\MT@set@all@ex#1{%
1280   debug\MT@debug@n1{3}{-- ex: setting all to \number#1}%
1281   \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1282 }
1283 \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.

```

1284 \MT@requires@pdftex4{
1285   \def\MT@reset@ef@codes{%
1286     \ifnum\MT@ex@factor@=\@m \else
1287       \MT@reset@ef@codes@
1288     \fi
1289   }
1290 }
1291 \let\MT@reset@ef@codes\MT@reset@ef@codes@

```



```

1292 }

\MT@ex@split@val    There's only one number per character.
1293 \def\MT@ex@split@val#1\relax{%
1294   \@tempcntb=#1\relax

   Take an optional factor into account.
1295   \ifnum\MT@ex@factor@=\@m \else
1296     \MT@scale\@tempcntb \MT@ex@factor@ \@m
1297   \fi
1298   \ifnum\@tempcntb > \MT@ex@max
1299     \MT@warn@ex@too@large\MT@ex@max
1300   \else
1301     \ifnum\@tempcntb < \MT@ex@min
1302       \MT@warn@ex@too@large\MT@ex@min
1303     \fi
1304   \fi
1305   \efcode\MT@font\MT@char=\@tempcntb
1306   (debug)\MT@edinfo@n{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%

   Heirs, heirs, I love thy heirs.
1307   \MT@ifdefined@c@T\MT@ex@inh@name{%
1308     \MT@ifdefined@n@T{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1309       \expandafter\MT@map@tlist@c
1310       \csname MT@inh@\MT@ex@inh@name @\MT@char @\endcsname
1311       \MT@set@ex@heirs
1312     }%
1313   }%
1314 }

\MT@warn@ex@too@large

1315 \def\MT@warn@ex@too@large#1{%
1316   \MT@warning@n{Expansion factor \number\@tempcntb\space too large for
1317     character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1318     Setting it to the maximum of \number#1}%
1319   \@tempcntb=#1\relax
1320 }

\MT@get@ex@opt    Apply different values to this font?
\MT@ex@factor@ 1321 \def\MT@get@ex@opt{%
\MT@stretch@ 1322   \MT@set@listname
1323   \MT@ifdefined@n@TF{\MT@ex@cc@\MT@ex@cc@name @factor}{%
\MT@shrink@ 1324     \MT@let@cn\MT@ex@factor@\MT@ex@cc@\MT@ex@cc@name @factor}%
\MT@step@ 1325     \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
1326   }%
\MT@auto@ 1327   \let\MT@ex@factor@\MT@ex@factor
1328   }%
1329   \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1330   \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1331   \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
1332   \def\@tempa{autoexpand}%
1333   \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1334   \MT@ifdefined@n@T{\MT@ex@cc@\MT@ex@cc@name @preset}{%
1335     \MT@preset@ex
1336     \let\MT@reset@ef@codes\relax
1337   }%
1338 }

\MT@get@ex@opt@

1339 \def\MT@get@ex@opt@#1#2{%
1340   \MT@ifdefined@n@TF{\MT@ex@cc@\MT@ex@cc@name @#1}{%
1341     \MT@let@nn{\MT@#10}{\MT@ex@cc@\MT@ex@cc@name @#1}%

```

```

1342     \MT@vinfo{... : #2}%
1343   }{%
1344     \MT@let@nn{MT@#1@}{MT@#1}%
1345   }%
1346 }

\MT@set@ex@heirs
1347 \def\MT@set@ex@heirs#1{%
1348   \efcode\MT@font#1=\efcode\MT@font\MT@char
1349   <debug>\MT@info{n1}{2}{-- heir of \MT@char: #1}%
1350   <debug>\MT@info{n1}{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1351 }

\MT@preset@ex
1352 \def\MT@preset@ex{%
1353   \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
1354   \MT@scale@factor
1355   \MT@set@all@ex\@tempcntb
1356 }

```

### 14.2.3 Interword Spacing (Glue)

```

\MT@spacing      Adjustment of interword spacing?
1357 \MT@requires@pdftex6{
1358   \def\MT@spacing{\MT@maybe@do{sp}}

\MT@set@sp@codes  This is all the same.
1359 \def\MT@set@sp@codes{%
1360   \MT@if@list@exists{%
1361     \MT@get@font@dimen@six
1362     \MT@get@opt
1363     \MT@reset@sp@codes
1364     \MT@get@inh@list
1365     \MT@set@inputenc{c}%
1366     \MT@load@list\MT@sp@c@name
1367     \MT@set@listname
1368     \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
1369     \expandafter\MT@set@codes\@tempc,\relax,%
1370   }\MT@reset@sp@codes
1371 }

\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).
1372 \def\MT@sp@split@val#1,#2,#3\relax{%
1373   \def\@tempb{#1}%
1374   \MT@ifempty\@tempb\relax{%
1375     \MT@get@space@unit2%
1376     \MT@scale@to@em
1377     \knbscode\MT@font\MT@char=\@tempcntb
1378     <debug>\MT@info{n1}{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1379   }%
1380   \def\@tempb{#2}%
1381   \MT@ifempty\@tempb\relax{%
1382     \MT@get@space@unit3%
1383     \MT@scale@to@em
1384     \stbscode\MT@font\MT@char=\@tempcntb
1385     <debug>\MT@info{n1}{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1386   }%
1387   \def\@tempb{#3}%
1388   \MT@ifempty\@tempb\relax{%

```

```

1389 \MT@get@space@unit4%
1390 \MT@scale@to@em
1391 \shbscode\MT@font\MT@char=\@tempcntb
1392 <debug>\MT@edinfo@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1393 }%
1394 \MT@ifdefined@c@T\MT@sp@inh@name{%
1395 \MT@ifdefined@n@T\MT@inh@\MT@sp@inh@name @\MT@char @}{%
1396 \expandafter\MT@map@tlist@c
1397 \csname MT@inh@\MT@sp@inh@name @\MT@char @\endcsname
1398 \MT@set@sp@heirs
1399 }%
1400 }%
1401 }

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

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

\MT@preset@sp
\MT@preset@sp@ 1420 \def\MT@preset@sp{%
1421 \expandafter\expandafter\expandafter\MT@preset@sp@
1422 \csname MT@sp@cc\MT@sp@cc@name @preset\endcsname\@nil
1423 }
1424 \def\MT@preset@sp@#1,#2,#3\@nil{%
1425 \ifx\MT@sp@unit@\@empty
1426 \MT@warn@preset@t@width{sp}%
1427 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
1428 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
1429 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
1430 \else
1431 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
1432 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
1433 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
1434 \fi
1435 \MT@set@all@sp\@tempa\@tempc\@tempb
1436 }
1437 }\relax

```

#### 14.2.4 Additional Kerning

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

```

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

```

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

```

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

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

```

1453 \def\MT@kn@split@val#1,#2\relax{%
1454   \def\@tempb{#1}%
1455   \MT@ifempty\@tempb\relax{%
1456     \MT@get@space@unit2%
1457     \MT@scale@to@em
1458     \knbcode\MT@font\MT@char=\@tempcntb
1459   <debug>\MT@info@n1{4}{;;; knbc (\MT@char): \number\knbcode\MT@font\MT@char: [#1]}%
1460   }%
1461   \def\@tempb{#2}%
1462   \MT@ifempty\@tempb\relax{%
1463     \MT@get@space@unit2%
1464     \MT@scale@to@em
1465     \knacode\MT@font\MT@char=\@tempcntb
1466   <debug>\MT@info@n1{4}{;;; knac (\MT@char): \number\knacode\MT@font\MT@char: [#2]}%
1467   }%
1468   \MT@ifdefined@c@T\MT@kn@inh@name{%
1469     \MT@ifdefined@nT\MT@inh@\MT@kn@inh@name @\MT@char @}%
1470     \expandafter\MT@map@tlist@c
1471     \csname MT@inh@\MT@kn@inh@name @\MT@char @\endcsname
1472     \MT@set@kn@heirs
1473   }%
1474   }%
1475 }
```

\MT@set@kn@heirs

```

1476 \def\MT@set@kn@heirs#1{%
1477   \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1478   \knacode\MT@font#1=\knacode\MT@font\MT@char
1479   <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1480   <debug>\MT@info@n1{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1481   <debug>                                     \number\knacode\MT@font\MT@char}%
1482 }
```

\MT@set@all@kn

```

\MT@reset@kn@codes 1483 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1484 <debug>\MT@info@n1{3}{-- knac/knbc: setting all to #1/#2}%
1485   \let\MT@temp@empty
1486   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbcode\MT@font\@tempcnta=#1\relax}}%
1487   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knacode\MT@font\@tempcnta=#2\relax}}%
1488   \MT@do@font\MT@temp
1489 }
1490 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1491 \let\MT@reset@kn@codes\relax
```

\MT@preset@kn

\MT@preset@kn@ 1492 \def\MT@preset@kn{%

```

1493 \expandafter\expandafter\expandafter\MT@preset@kn@
1494 \csname MT@kn@c@MT@kn@c@name @preset\endcsname\@nil
1495 }
1496 \def\MT@preset@kn@#1,#2\@nil{%
1497 \ifx\MT@kn@unit@\empty
1498 \MT@warn@preset@tewidth{kn}%
1499 \let\MT@preset@aux\MT@preset@aux@factor
1500 \else
1501 \def\MT@preset@aux{\MT@preset@aux@space2}%
1502 \fi
1503 \MT@ifempty{#1}{\let\@tempa\empty}{\MT@preset@aux{#1}\@tempa}%
1504 \MT@ifempty{#2}{\let\@tempb\empty}{\MT@preset@aux{#2}\@tempb}%
1505 \MT@set@all@kn\@tempa\@tempb
1506 }
1507 }\relax

```

### 14.2.5 Tracking

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

```
1508 \MT@requires@pdftex{
```

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

```

\MT@tr@font@list 1509 \let\MT@tr@font@list\empty
1510 \def\MT@tracking@{%
1511 \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
1512 \ifMT@inlist\else
1513 \MT@maybe@do{tr}%
1514 \ifMT@do\else
1515 \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1516 \fi
1517 \fi
1518 }
1519 \let\MT@tracking\MT@tracking@
1520 </package>
1521 <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.

```

1522 \def\MT@set@tr@codes{%
1523 <*package>
1524 \MT@vinfo{Tracking font ~\MT@font'\on@line}%
1525 \MT@get@font@dimen@six
1526 \MT@if@list@exists
1527 \MT@get@tr@opt
1528 \relax
1529 </package>
1530 \MT@ifdefined@c@TF\MT@letterspace@relax{\let\MT@letterspace@\MT@letterspace}%
1531 \ifnum\MT@letterspace@=\z@else
1532 <package> \MT@vinfo{... Tracking by \number\MT@letterspace@}%

```

Letterspacing only works in PDF mode.

```
1533 \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.

```

1534 \xdef\MT@lsfont{\csname\expandafter\string\font@name
1535 \number\MT@letterspace@ls\endcsname}%
1536 \expandafter\ifx\MT@lsfont\relax
1537 <debug>\MT@info@n1{1}{... new letterspacing instance}%

```

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

```

1538 \MT@get@ls@basefont
1539 \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@

```

Scale interword spacing (not configurable in letterspace).

```

1540 <*package>
1541 \MT@ifdefined@c@TF\MT@tr@ispace
1542 {\let\@tempa\MT@tr@ispace}%
1543 {\edef\@tempa{\MT@letterspace@*,,}}%
1544 \MT@ifdefined@c@TF\MT@tr@ospace
1545 {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1546 {\edef\@tempa{\@tempa,,,}}%
1547 \expandafter\MT@tr@set@space\@tempa,%

```

Adjust outer kerning.

```

1548 \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
1549 \expandafter\MT@tr@set@okern\@tempa,%
1550 </package>
1551 <*letterspace>
1552 \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1553 * \fontdimen2\MT@lsfont/1000\relax
1554 </letterspace>

```

Disable ligatures (not configurable in letterspace).

```

1555 <package> \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1556 <letterspace> \tagcode\MT@lsfont`f=\m@ne
1557 \fi

```

Finally, let the letterspaced font propagate.

```

1558 \aftergroup\MT@set@lsfont
1559 <package> \let\MT@font\MT@lsfont

```

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

```

\MT@curr@ls 1560 \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
1561 \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.

```

1562 <*package>
1563 \MT@outer@space=\csname \MT@outer@space\expandafter\string\font@name\endcsname\relax
1564 \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1565 \aftergroup\aftergroup\aftergroup\MT@set@curr@os
1566 \MT@tr@outer@l
1567 </package>

```

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

```

1568 \ifx\MT@ls@adjust\@empty
1569 \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1570 \MT@ls@outer@k
1571 <*letterspace>
1572 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1573 \aftergroup\aftergroup\aftergroup\MT@set@curr@ok
1574 \aftergroup\aftergroup\aftergroup\MT@ls@outer@k

```

1575 *(/letterspace)*

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

```

1576 (*package)
1577   \else
1578     \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
1579       \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1580     \ifdim\MT@outer@kern=z@ \else \MT@ls@outer@k \fi
1581     \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
1582       \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1583 (/package)
1584 \fi
1585 (*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).

```

1586 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1587 \aftergroup\aftergroup\aftergroup\MT@set@curr@ok
1588 \aftergroup\aftergroup\aftergroup\MT@tr@outer@r

```

Adjust protrusion values, now and maybe later.

```

1589 (debug)\MT@info{n}{2}{... compensating for tracking (\number\MT@letterspace@)}%
1590 (/package)
1591 \MT@do@font{\lcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
1592 \rprcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
1593 (package) \let\MT@the@prcode\MT@the@prcode@tr
1594 \fi
1595 }

```

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

```

1596 (*package)
1597 \def\MT@get@tr@opt{%
1598   \MT@set@listname
1599   \MT@ifdefined@n@T{\MT@tr@c@\MT@tr@c@name}{%
1600     \MT@let@cn\MT@letterspace{\MT@tr@c@\MT@tr@c@name}%

```

**\MT@tr@unit@** Different unit?

```

1601 \MT@ifdefined@n@T{\MT@tr@c@\MT@tr@c@name @unit}{%
1602 \MT@let@cn\MT@tr@unit@{\MT@tr@c@\MT@tr@c@name @unit}%
1603 \ifdim\MT@tr@unit@=1em
1604 \let\MT@tr@unit@\undefined
1605 \else
1606 \MT@let@cn\@tempb{\MT@tr@c@\MT@tr@c@name}%
1607 \MT@get@unit\MT@tr@unit@
1608 \let\MT@tr@factor@\@m
1609 \MT@scale@to@em
1610 \edef\MT@letterspace{\number\@tempcntb}%
1611 \fi
1612 }%
1613 }%

```

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

```

\MT@tr@ospace 1614 \MT@get@tr@opt@{spacing} {ispace}%
1615 \MT@get@tr@opt@{outerspace}{ospace}%

```

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

```

1616 \MT@get@tr@opt@{outerkerning}{okern}%

```

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

```

1617 \MT@get@tr@opt@{noligatures} {ligatures}%
1618 }

```

`\MT@get@tr@opt@`

```
1619 \def\MT@get@tr@opt@#1#2{%
1620   \MT@ifdefined@n@T{\MT@tr@c@{\MT@tr@c@name @#1}%
1621     {\MT@let@nn{\MT@tr@#2}{\MT@tr@c@{\MT@tr@c@name @#1}}}%
1622 }
1623 </package>
```

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

```
1624 <plain>\ifnum\MT@plain=0
1625 \def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}
```

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

```
1626 \DeclareRobustCommand\lsstyle{%
1627   <package> \def\MT@feat{tr}%
1628   \let\MT@tracking\MT@set@tr@codes
1629   \selectfont
1630 }
```

Now the definitions for the letterspace package with plain  $\text{\TeX}$ .

```
1631 <*plain>
1632 \else
1633 \def\MT@set@lsfont{\MT@lsfont}
1634 \def\lsstyle{%
1635   \begingroup
1636   \escapechar\m@ne
1637   \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1638   \MT@set@tr@codes
1639   \endgroup
1640 }
1641 \let\textls\undefined
1642 \let\lslig\undefined
1643 \fi
1644 </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.

```
\MT@lslig
1645 \DeclareRobustCommand\lslig[1]{%
1646   {\MT@ifdefined@c@TF\MT@curr@ls{%
1647     \escapechar\m@ne
1648     \MT@get@ls@basefont
1649     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1650     \kern\MT@outer@kern
1651     \font@name #1%
1652     \kern\MT@outer@kern%
1653   }{#1}}%
1654 }
```

`\MT@get@ls@basefont` pdf $\text{\TeX}$  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.

```
1655 \def\MT@get@ls@basefont{%
1656   \edef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1657   \expandafter\ifx\MT@ls@basefont\relax
1658     \MT@exp@two@c\MT@gl@et\MT@ls@basefont\font@name
1659   \else
1660     <debug>\MT@info@n1{1}{... fixing base font}%
1661     \MT@exp@two@c\let\font@name\MT@ls@basefont
1662   \fi
```



```

1663 }

\MT@tr@noligatures pdfTeX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.
1664 <package>
1665 \MT@requires@pdftex7{
1666   \def\MT@tr@noligatures{%
1667     \ifx\MT@tr@ligatures\@empty
1668       \MT@noligatures@\MT@lsfont\@undefined
1669     \else
1670       \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1671     \fi
1672   }
1673 }{
1674   \def\MT@tr@noligatures{%
1675     \MT@warning@n1{%
1676       Disabling selected ligatures is only possible since\MessageBreak
1677       pdftex 1.40.4. Disabling all ligatures instead}%
1678     \global\let\MT@tr@noligatures\relax
1679   }
1680 }

\MT@outer@space A new skip for outer spacing.
1681 \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.
1682 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
1683 <debug>\MT@info@n12{... orig. space: \the\fontdimen2\MT@lsfont,
1684 <debug> \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
1685 <debug> \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
1686   \let\MT@temp\@empty
1687   \MT@tr@set@space@{#1}{#4}{2}\@empty
1688   \MT@tr@set@space@{#2}{#5}{3}\@plus
1689   \MT@tr@set@space@{#3}{#6}{4}\@minus
1690   \global\MT@let@nc{\MT@outer@space\expandafter\string\font@name}\MT@temp
1691 <debug>\MT@info@n12{... inner space: \the\fontdimen2\MT@lsfont,
1692 <debug> \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
1693 <debug>\MT@info@n12{... outer space: \MT@temp}%
1694 }

\MT@tr@set@space@ If outer spacing settings don't exist, they will be inherited from the inner spacing
settings.
1695 \def\MT@tr@set@space@#1#2#3#4{%
1696   \MT@ifempty{#2}{%
1697     \MT@ifempty{#1}{%
1698       \edef\MT@temp{\MT@temp#4\the\fontdimen3\MT@lsfont}%
1699     }{%
1700       \MT@tr@set@space@@{#1}{#3}{1000}%
1701       \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1702       \fontdimen3\MT@lsfont=\@tempdima
1703     }%
1704   }{%
1705     \MT@tr@set@space@@{#2}{#3}{2000}%
1706     \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1707     \MT@ifempty{#1}{\relax}%
1708     \MT@tr@set@space@@{#1}{#3}{1000}%
1709     \fontdimen3\MT@lsfont=\@tempdima
1710   }%
1711 }%
1712 }

```

`\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.

```

1713 \def\MT@tr@set@space@#1#2#3{%
1714   \MT@test@ast#1*\@nil{%
1715     \MT@ifdefined@c@TF\MT@tr@unit@
1716     {\edef\@tempb{#1}\MT@scale@to@em}
1717     {\@tempcntb=#1\relax}%
1718     \@tempdima=\dimexpr\dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
1719     -\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).

```

1720   \ifnum#2=\tw@
1721   \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
1722   \fi
1723   \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
1724   }{%
1725   \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
1726   \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
1727   }%
1728   (debug)\MT@info@n13{... : font dimen #2 (#1): \the\@tempdima}%
1729 }

```

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

```

1730 \def\MT@tr@outer@{%
1731   \ifdim\lastskip>5sp
1732   \edef\x{\the\lastskip plus 0pt}%
1733   \setbox\z@\hbox{\MT@outer@space=\x}%
1734   \ifdim\wd\z@>\z@
1735   (debug)\MT@info2{[[[ adjusting pre spaces: \the\MT@outer@space}%
1736   \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

1737   \let\MT@ls@outer@k\relax
1738   \fi
1739   \fi
1740 }

```

`\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 1741 \def\MT@tr@outer@r{%
1742   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
1743 }
1744 \def\MT@tr@outer@r@{%
1745   \def\MT@temp*#1{\hskip\MT@outer@space
1746   (debug)\MT@info2{[[[ adjusting post spaces (1): \the\MT@outer@space}%
1747   }%
1748   \ifcat\egroup\noexpand\MT@tr@outer@next
1749   \unkern\egroup
1750   \MT@set@curr@ok
1751   \MT@set@curr@os
1752   \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\x=}%
1753   \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.

```

1754   \ifx\maybe@ic\MT@tr@outer@next
1755   \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\x=}%

```

1756     \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).

```

1757       \ifx\check@icr\MT@tr@outer@next
1758       \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\x=%}
1759       \else
1760       \ifx\@sptoken\MT@tr@outer@next
1761       \def\MT@temp* {\hskip\MT@outer@space
1762 <debug>\MT@edinfo2{}}] adjusting post spaces (2): \the\MT@outer@space}%
1763       }%
1764       \else
1765       \ifx-\MT@tr@outer@next
1766       \def\MT@temp*~{\nobreak\hskip\MT@outer@space
1767 <debug>\MT@edinfo2{}}] adjusting post spaces (3): \the\MT@outer@space}%
1768       }%
1769       \else
1770       \ifx\ \MT@tr@outer@next \else
1771       \ifx\space\MT@tr@outer@next \else
1772       \ifx\@xobeysp\MT@tr@outer@next \else

```

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

```

1773               \def\MT@temp*{\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k\fi}%
1774               \let\MT@tr@outer@next\relax
1775       \fi\fi\fi\fi\fi\fi\fi\fi
1776       \MT@temp*%
1777 }

```

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

```

\MT@tr@outer@icr@ 1778 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
1779 \def\MT@tr@outer@icr@{%
1780   \let\@let@token= \MT@tr@outer@next
1781   \maybe@ic@
1782 }

```

For older pdfTeX versions, throw an error.

```

1783 }{
1784   \DeclareRobustCommand\lsstyle{%
1785   \MT@error{Letterspacing only works with pdfTeX version 1.40\MessageBreak
1786   or newer}{Upgrade pdfTeX, or use the 'soul' package instead.}%
1787   \MT@glet\lsstyle\relax
1788 }
1789 }
1790 </package>

```

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

```

1791 \DeclareRobustCommand\textls{%
1792   \hmode\bgroup
1793   \@ifstar{\let\MT@ls@adjust\@empty\MT@textls}%
1794   {\let\MT@ls@adjust\relax \MT@textls}%
1795 }

```

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

```

\MT@letterspace@ 1796 \newcommand\MT@textls[2][\]{%
1797   \MT@ifempty{#1}%
1798   {\let\MT@letterspace@\@undefined}%
1799   {\KV@esp@def\MT@letterspace@{#1}%
1800    \MT@ls@too@large\MT@letterspace@}%
1801   \lsstyle #2%
1802   \egroup
1803 }

```

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

```

1804 \def\MT@ls@too@large#1{%
1805   \ifnum#1>\MT@tr@max
1806     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
1807     \let#1\MT@tr@max
1808   \else
1809     \ifnum#1<\MT@tr@min
1810       \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
1811       \let#1\MT@tr@min
1812     \fi
1813   \fi
1814 }

\MT@outer@kern      This dimen is used for the starred version of \textls, for \lslig and for adjusted
\MT@tr@set@okern      outer kerning.
1815 \newdimen\MT@outer@kern
1816 <package>
1817 \def\MT@tr@set@okern#1,#2,{%
1818   \let\MT@temp@empty
1819   \MT@tr@set@okern@{#1}%
1820   \MT@tr@set@okern@{#2}%
1821   \global\MT@let@nc\MT@outer@kern\expandafter\string\font@name}\MT@temp
1822   <debug>\MT@dinfnl2{... outer kerning:
1823   <debug>      (#1,#2) = \nameuse{MT@outer@kern\expandafter\string\font@name}}%
1824 }

\MT@tr@set@okern@
1825 \def\MT@tr@set@okern@#1{%
1826   \MT@test@ast#1*\@nil{%
1827     \MT@ifdefined@cTF\MT@tr@unit@
1828     {\edef\@tempb{#1}\MT@scale@to@em}
1829     {\@tempcntb=#1\relax}%
1830     \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
1831   }{%
1832     \MT@ifempty\@tempa{\let\@tempa\@m}\relax
1833     \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
1834     * \fontdimen6\MT@lsfont/2000\relax
1835   }%
1836   \advance\@tempdima -\dimexpr \MT@letterspace@ sp
1837   * \fontdimen6\MT@lsfont/2000\relax
1838   \edef\MT@temp{\MT@temp{\the\@tempdima}}%
1839 }
1840 </package>

\MT@ls@outer@k      Adjust outer kerning.
1841 \def\MT@ls@outer@k{\ifhmode\kern\MT@outer@kern\relax\fi}
1842 <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.

```

1843 \MT@requires@pdftex5{
1844 \def\MT@noligatures{%
1845   \MT@dotrue
1846   \let\@tempa\MT@nl@setname
1847   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1848     \MT@ifdefined@cTF{MT@checklist@##1}%
1849     {\csname MT@checklist@##1\endcsname}%
1850     {\MT@checklist@{##1}}}%
1851   {nl}%

```

```

1852 }%
1853 \ifMT@do
1854   \MT@noligatures@MT@font\MT@n1@ligatures
1855 \fi
1856 }

```

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

```

1857 \def\MT@noligatures@#1#2{%
1858   \MT@ifdefined@c@TF#2{%

```

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

```

1859   \MT@ifdefined@c@TF\tagcode{%
1860     \MT@map@c@list@c#2{%
1861       \KV@sp@def\@tempa{##1}\MT@get@slot
1862       \ifnum\MT@char > \m@ne
1863         \tagcode#1\MT@char=\m@ne\fi
1864     }%
1865     \MT@vinfo{... Disabling ligatures for characters: #2}%
1866   }{%
1867     \pdfnoligatures#1%
1868     \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
1869       know \backslashchar \tagcode). Disabling all ligatures of\MessageBreak
1870       the font instead}%
1871   }%
1872 }{%
1873   \pdfnoligatures#1%
1874   \MT@vinfo{... Disabling ligatures}%
1875 }%
1876 }
1877 }\relax

```

### 14.2.7 Loading the Configuration

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

```

1878 \def\MT@load@list#1{%
1879   \edef\@tempa{#1}%
1880   \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa @load}%
1881   \MT@ifstreq\@tempa\@tempb{%
1882     \MT@error{\@nameuse{MT@abbr@MT@feat} list \@tempa' cannot load itself}{}%
1883   }{%
1884     \ifx\@tempb\relax \else
1885       \MT@ifdefined@n@TF{MT@MT@feat @c@\@tempb}{%
1886         \MT@vinfo{... : First loading \@nameuse{MT@abbr@MT@feat} list \@tempb'}%
1887         \begingroup
1888           \MT@load@list\@tempb
1889         \endgroup
1890         \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list
1891           \noexpand\MessageBreak \@tempb'}%
1892         \MT@let@cn\@tempc{MT@MT@feat @c@\@tempb}%
1893         \expandafter\MT@set@codes\@tempc,\relax,%
1894       }{%
1895         \MT@error{\@nameuse{MT@abbr@MT@feat} list \@tempb' undefined.\MessageBreak
1896           Cannot load it from list \@tempa'}{}%
1897       }%
1898     \fi
1899   }%
1900 }

```

\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.

```

1901 \let\MT@file@list\@empty
1902 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

1903 \MT@in@clist{#1}\MT@file@list
1904 \ifMT@inlist@ \else

```

Don't forget that because reading the files takes place inside a group, all commands that may be used there have to be defined globally.

```

1905 \MT@begin@catcodes
1906 \let\MT@begin@catcodes\relax
1907 \let\MT@end@catcodes\relax
1908 \InputIfFileExists{mt-#1.cfg}{%
1909 \edef\MT@curr@file{mt-#1.cfg}%
1910 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1911 \MT@xadd\MT@file@list{#1,}%
1912 }{%
1913 \MT@get@basefamily#1\@empty\@empty\@empty\@nil
1914 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1915 \ifMT@inlist@
1916 \MT@xadd\MT@file@list{#1,}%
1917 \else
1918 \InputIfFileExists{mt-\@tempa.cfg}{%
1919 \edef\MT@curr@file{mt-\@tempa.cfg}%
1920 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1921 \MT@xadd\MT@file@list{\@tempa,#1,}%
1922 }{%
1923 \MT@vinfo{... No configuration file mt-#1.cfg}%
1924 \MT@xadd\MT@file@list{#1,}%
1925 }%
1926 \fi
1927 }%
1928 \endgroup
1929 \fi
1930 }

```

\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.

```

1931 \def\MT@cfg@catcodes{%
1932 \makeatletter
1933 \catcode\^7%
1934 \catcode\ 9%
1935 \catcode\^^I9%
1936 \catcode\^^M9%
1937 \catcode\\\z@
1938 \catcode\{\@one
1939 \catcode\}\@tw@
1940 \catcode\#6%
1941 \catcode\%14%
1942 \MT@map@tlist@n
1943 {\!\"$&'\"(\)*\+,\,-\.\./\:\;\<=\>?\"[\]\_\"|\|-}%
1944 \makeother
1945 }

```

\MT@begin@catcodes

This will be used before reading the files as well as in the configuration com-

Table 4: Order for matching font attributes

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

mands `\Set...`, and `\DeclareCharacterInheritance`, so that the catcodes are also harmless when these commands are used outside the configuration files.

```

1946 \def\MT@begin@catcodes{%
1947   \begingroup
1948   \MT@cfg@catcodes
1949 }
```

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

```

1950 \let\MT@end@catcodes\endgroup
```

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

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

```

1951 \def\MT@get@basefamily#1#2#3#4\@nil{%
1952   \ifx\@empty#4%
1953     \def\@tempa{#1#2#3}%
1954   \else
1955     \let\@tempa\@empty
1956     \edef\@tempb{#1#2#3#4}%
1957     \expandafter\MT@get@basefamily@\@tempb\@nil
1958   \fi
1959 }
```

`\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'.

```

1960 \def\MT@get@basefamily@#1#2\@nil{%
1961   \edef\@tempa{\@tempa#1}%
1962   \ifx\@#2\@expandafter\@gobble\else\expandafter\@firstofone\fi
1963   {\MT@in@tlist{#2}\MT@variants
1964    \ifMT@inlist\else\MT@get@basefamily@#2\@nil\fi}%
1965 }
```

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

`\MT@get@listname`

```

1966 \def\MT@get@listname#1{%
1967   (debug)\MT@info@n1{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
1968   \let\MT@listname\@undefined
1969   \def\@tempb{#1}%
1970   \MT@map@tlist@c\MT@try@order\MT@get@listname@
1971 }
1972 \def\MT@get@listname@#1{%
1973   \expandafter\MT@next@listname#1%
1974   \ifx\MT@listname\@undefined \else
1975     \expandafter\MT@tlist@break
1976   \fi
1977 }
```

`\MT@try@order` Beginning with version 1.7, we always check for the font size. Since the matching order has become more logical now, it can be described in words, so that we don't need table 4 in the documentation part any longer and can cast it off here.

```
1978 \def\MT@try@order{%
1979   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
1980   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
1981 }
```

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

```
1982 \def\MT@next@listname#1#2#3#4{%
1983   \edef\@tempa{\MT@encoding
1984     /\ifnum#1=\@ne \MT@family\fi
1985     /\ifnum#2=\@ne \MT@series\fi
1986     /\ifnum#3=\@ne \MT@shape\fi
1987     /\ifnum#4=\@ne *\fi
1988     \MT@context}%
1989   (debug)\MT@info{n}{1}{trying \@tempa}%
1990   \MT@ifdefined@nTF{MT@\@tempb @\@tempa}{%
1991     \MT@next@listname@#4%
1992   }%
```

Also try with an alias family.

```
1993   \ifnum#1=\@ne
1994     \ifx\MT@familyalias\empty \else
1995       \edef\@tempa{\MT@encoding
1996         /\MT@familyalias
1997         /\ifnum#2=\@ne \MT@series\fi
1998         /\ifnum#3=\@ne \MT@shape\fi
1999         /\ifnum#4=\@ne *\fi
2000         \MT@context}%
2001     (debug)\MT@info{n}{1}{(alias) \@tempa}%
2002     \MT@ifdefined@nTF{MT@\@tempb @\@tempa}{%
2003       \MT@next@listname@#4%
2004     }%
2005   \fi
2006 \fi
2007 }%
2008 }
```

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

```
2009 \def\MT@next@listname@#1{%
2010   \ifnum#1=\@ne
2011     \expandafter\MT@in@rlist\csname MT@\@tempb @\@tempa @sizes\endcsname
2012     \ifMT@inlist@
2013       \let\MT@listname\MT@size@name
2014     \fi
2015   \else
2016     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
2017   \fi
2018 }
```

`\MT@if@list@exists`

```
\MT@context 2019 \def\MT@if@list@exists{%
2020   \MT@let@cn\MT@context{MT@\MT@feat @context}%
2021   \MT@ifstreq{@}\MT@context{\let\MT@context\empty}\relax
2022   \MT@get@listname{\MT@feat @c}%
2023   \MT@ifdefined@cTF\MT@listname{%
2024     \MT@edef{n}{MT@\MT@feat @c@name}{\MT@listname}%
2025     \ifMT@nonselected
2026       \MT@vinfo{... Applying non-selected expansion (list '\MT@listname')}%
2027     \else
```



```

2028     \MT@vinfo{... Loading \@nameuse{MT@abbr@MT@feat} list `\'MT@listname'}%
2029     \fi
2030     \@firstoftwo
2031 }{%

```

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

```

2032     \MT@let@nc{MT@MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

2033     \ifMT@nonselected
2034     \MT@vinfo{... Applying non-selected expansion (no list)}%
2035     \else

```

Tracking doesn't require a list, either.

```

2036     \MT@ifstreq{MT@feat}{tr}\relax{%
2037     \MT@warning{I cannot find a \@nameuse{MT@abbr@MT@feat} list
2038         for font\MessageBreak`\'MT@font'%
2039         \ifx\MT@context\@empty\else\space(context: `\'MT@context')\fi.
2040         Switching off\MessageBreak\@nameuse{MT@abbr@MT@feat} for this font}%
2041     }%
2042     \fi
2043     \@secondoftwo
2044 }%
2045 }

```

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

```

\MT@context 2046 \def\MT@get@inh@list{%
2047     \let\MT@context\@empty
2048     \MT@get@listname{\MT@feat @inh}%
2049     \MT@ifdefined@c@TF\MT@listname{%
2050         \MT@edef@n{MT@MT@feat @inh@name}{\'MT@listname}%
2051         <debug>\MT@dinfo@n{1}{... Using \@nameuse{MT@abbr@MT@feat} inheritance list
2052         <debug>         `\'MT@listname'}%
2053         \MT@let@cn\@tempc{MT@MT@feat @inh@MT@listname}%

```

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

```

2054     \ifx\@tempc\@empty \else
2055     <debug>\MT@dinfo@n{1}{parsing inheritance list ...}%

```

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

```

2056     \begingroup
2057     \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\'MT@listname'}%
2058     \MT@set@inputenc{inh}%
2059     \expandafter\MT@inh@do\@tempc,\relax,%
2060     \global\MT@let@nc{MT@MT@feat @inh@MT@listname}\@empty
2061     \endgroup
2062     \fi
2063 }{%
2064     \MT@let@nc{MT@MT@feat @inh@name}\@undefined
2065 }%
2066 }

```

#### 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@ 2067 \def\MT@get@slot{%
2068   \escapechar~\
2069   \let\MT@char@m@ne
2070   \MT@noesttrue
```

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

```
2071   \MT@toks=\expandafter{\@tempa}%
```

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

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

```
2072   \expandafter\MT@is@letter\@tempa\relax\relax
2073   \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.

```
2074   \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.

```
2075   \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
2076   \MT@is@symbol
```

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

```
2077   {\expandafter\MT@is@composite\@tempa\relax\relax}%
2078   \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.

```
2079   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2080   \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2081   \fi
2082   \fi

2083   \let\MT@char\MT@char@
2084   \ifnum\MT@char < \z@
2085     \MT@warn@unknown
2086   \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.

```
2087   \ifMT@noest \else
2088     \MT@warn@rest
2089     \let\MT@char@m@ne
2090     \fi
2091   \fi
```

```

2092 \escapechar\m@ne
2093 }

\ifMT@norest    Test whether all of the string has been used up.
2094 \newif\ifMT@norest

\MT@is@letter    Input is a letter, a character or a number.
2095 \def\MT@is@letter#1#2\relax{%
2096   \ifcat a\noexpand#1\relax
2097     \edef\MT@char@{\number`#1}%
2098     \ifx\#2\%
2099       (debug)\MT@info@n1{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2100     \else
2101       \MT@norestfalse
2102     \fi
2103   \else
2104     \ifcat !\noexpand#1\relax
2105       \edef\MT@char@{\number`#1}%
2106       (debug)\MT@info@n1{3}{> `the\MT@toks' is a character (\MT@char@)}%
2107       \ifx\#2\%
2108         \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
2109       \else
2110         \MT@norestfalse
2111         \expandafter\MT@is@number#1#2\relax\relax
2112       \fi
2113     \fi
2114   \fi
2115 }

\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.
2116 \def\MT@is@number#1#2#3\relax{%
2117   \ifx\relax#3\relax \else
2118     \ifx\relax#2\relax \else
2119       \MT@noresttrue
2120       \if#1"\relax
2121         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2122       (debug)\MT@info@n1{3}{> ... a hexadecimal number: \MT@char@}%
2123     \else
2124       \if#1'\relax
2125         \def\MT@char@{\number#1#2#3}%
2126       (debug)\MT@info@n1{3}{> ... an octal number: \MT@char@}%
2127     \else
2128       \MT@ifint{#1#2#3}{%
2129         \def\MT@char@{\number#1#2#3}%
2130       (debug)\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
2131       }\MT@norestfalse
2132     \fi
2133   \fi
2134   \ifnum\MT@char@ > \cc1v
2135     \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2136     \let\MT@char@\m@ne
2137   \fi
2138 \fi
2139 \fi
2140 }

\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.

```
2141 \def\MT@is@active#1#2\@nil{%
2142   \ifnum\catcode`#1 = \active
2143     \begingroup
2144     \set@display@protect
2145     \let\IeC\@firstofone
2146     \let\@inpenc@undefined@MT@undefined@char
```

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

```
2147   \def\UTFviii@defined##1{\ifx ##1\relax
2148     \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
```

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

```
2149   \MT@ifdefined@c@T\PrerenderUnicode
2150   {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
2151   \edef\x{\endgroup
2152     \def\noexpand\@tempa{\@tempa}%
```

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

```
2153   \MT@toks={\the\MT@toks\space(= \@tempa)}%
2154   }%
2155   \x
2156   \fi
2157 }
```

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

```
2158 \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.

```
2159 \def\MT@is@symbol{%
2160   \expandafter\def\expandafter\MT@char\expandafter
2161     {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2162   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2163     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2164   \ifnum\MT@char@ < \z@
```

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

```
2165   \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2166   \fi
2167 }
```

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

```
\MT@charstring 2168 \begingroup
2169   \catcode`\=/\z@

2170   /MT@map@tlist@n{\CHAR}/@makeoother
2171   /lowercase{%
2172     /def/x{/endgroup
2173     /def/MT@charstring{\CHAR}%
```

```

2174 /def\MT@is@char##1\CHAR"##2##3##4/relax{%
2175 /ifx/relax##1/relax
2176 /if##3\relax
2177 /edef\MT@char@{/number"##2}%
2178 /MT@ifstreq\MT@charstring{##3##4}/relax/MT@noestfalse
2179 /else
2180 /edef\MT@char@{/number"##2##3}%
2181 /MT@ifstreq\MT@charstring{##4}/relax/MT@noestfalse
2182 /fi
2183 (debug) /MT@dinfo@n1{3}{> `the/MT@toks' is a \char (/MT@char@)}%
2184 /fi
2185 }%
2186 }%
2187 }
2188 /x

```

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

```

2189 \def\MT@is@composite#1#2\relax{%
2190 \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 `\stringify`  
it. Thus, we will die gracefully even on wrong Unicode input without `utf8`.

```

2191 \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2192 \string\csname\MT@encoding\endcsname
2193 \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
2194 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2195 \fi
2196 }

```

[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 2197 \def\MT@set@listname{%
2198 \edef\MT@curr@list@name{\@nameuse{MT@abbr@}\MT@feat} list\noexpand\MessageBreak
2199 ~\@nameuse{MT@\MT@feat @c@name}'}%
2200 }

```

```

\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.
2201 \def\MT@warn@ascii{%
2202   \MT@warning@nl{Character ‘\the\MT@toks’ (= \MT@char@)
2203     is outside of ASCII range.\MessageBreak
2204     You must load the ‘inputenc’ package before using\MessageBreak
2205     8-bit characters in \MT@curr@list@name}%
2206 }

\MT@warn@number@too@large    Number too large.
2207 \def\MT@warn@number@too@large#1{%
2208   \MT@warning@nl{%
2209     Number #1 in encoding ‘\MT@encoding’ too large!\MessageBreak
2210     Ignoring it in \MT@curr@list@name}%
2211 }

\MT@warn@rest      Not all of the string has been parsed.
2212 \def\MT@warn@rest{%
2213   \MT@warning@nl{%
2214     Unknown slot number of character\MessageBreak‘\the\MT@toks’%
2215     \MT@warn@maybe@inputenc\MessageBreak
2216     in font encoding ‘\MT@encoding’.\MessageBreak
2217     Make sure it’s a single character\MessageBreak
2218     (or a number) in \MT@curr@list@name}%
2219 }

\MT@warn@unknown   No idea what went wrong.
2220 \def\MT@warn@unknown{%
2221   \MT@warning@nl{%
2222     Unknown slot number of character\MessageBreak‘\the\MT@toks’%
2223     \MT@warn@maybe@inputenc\MessageBreak
2224     in font encoding ‘\MT@encoding’ in \MT@curr@list@name}%
2225 }

\MT@warn@maybe@inputenc    In case an input encoding had been requested.
2226 \def\MT@warn@maybe@inputenc{%
2227   \MT@ifdefined@n@T
2228   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
2229   { (input encoding ‘\@nameuse
2230     {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}’)}%
2231 }

```

## 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 my reference:

- `\pickup@font` is called by `\selectfont`, `\wrong@fontshape`, or `\getanddefine@fonts` (for math).
- `\pickup@font` calls `\define@newfont`.

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

Up to version 1.3 of this package, we were using `\define@newfont` as the hook, which is only called for *new* fonts, and therefore seemed the natural choice. However, this meant that we had to take special care to catch all fonts: we additionally had to set up the default font, the error font (if it wasn't the default font), we had to check for some packages that might have been loaded before `microtype` and were loading fonts, e. g., `jurabib`, `ledmac`, `pi font` (loaded by `hyperref`), `tipa`, and probably many more. Furthermore, we had to include a hack for the `IEEEtran` class which loads all fonts in the class file itself (to fine tune inter-word spacing). 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 2232 \let\MT@font@list\empty
2233 \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`.

```
2234 \package
2235 \plain\ifnum\MT@plain=0
2236 \MT@addto@setup%
```

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

```
2237 \ifpackageloaded{CJK}{
2238   \ifpackageafter{CJK}{2006/10/17}
2239   {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}} % 4.7.0
2240   {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}}}}
2241   \g@addto@macro\MT@orig@pickupfont{%
2242     {\expandafter\ifx\font@name\relax\define@newfont\fi}%
2243     {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2244       \define@newfont\else\xdef\font@name
2245         {\csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}
2246   }{
2247   \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
2248   }
```

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

```
2249 \ifx\pickup@font\MT@orig@pickupfont \else
2250   \MT@warning@n1{%
2251     Command \string\pickup@font\space is not defined as expected.%
2252     \MessageBreak Patching it anyway. Some things may break%
2253   }(*package)
2254   .\MessageBreak Double-check whether micro-typography is indeed%
2255   \MessageBreak applied to the document.%
2256   \MessageBreak (Hint: Turn on `verbose' mode)%
```

```

2257 </package>
2258 }
2259 \fi

```

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

```

2260 \g@addto@macro\pickup@font{\begingroup}

```

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

```

2261 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
2262 \g@addto@macro\pickup@font{%
2263   \escapechar\m@ne
2264 <*package>
2265 <debug>   \global\MT@inannottrue
2266 <debug>   \MT@glet\MT@pdf@annot\@empty
2267 <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.

```

2268   \MT@let@cn\MT@font{MT@subst\expandafter\string\font@name}%
2269   \ifx\MT@font\relax
2270     \let\MT@font\font@name
2271   \else
2272     \ifx\MT@font\font@name \else
2273 <debug> \MT@addto@annot{= substituted with \MT@font}%
2274     \MT@register@subst@font
2275   \fi
2276 \fi
2277 \MT@setupfont
2278 </package>
2279 <letterspace> \MT@tracking
2280 \endgroup
2281 }
2282 <*package>

```

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

```

2283 \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.

```

2284 \g@addto@macro\do@subst@correction
2285 { \edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2286   \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` would break the accent if different fonts are used for the base character and the accent. Fortunately,  $\text{\LaTeX}$  takes care that the fonts used for the `\accent` are already set up, so that we cannot be overlooking them.

```

2287 \let\MT@orig@add@accent\add@accent
2288 \def\add@accent#1#2{%
2289   \let\pickup@font\MT@orig@pickupfont
2290   \MT@orig@add@accent{#1}{#2}%
2291   \let\pickup@font\MT@pickupfont
2292 }

```



```

2293 </package>
2294 }
2295 <plain>\fi
2296 <*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.

```
2297 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

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

```
2298 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
```

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

```
2299 \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.

```
2300 \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.

```

2301 \def\MT@check@font@cx{%
2302   \MT@if@true
2303   \MT@map@clist@c\MT@active@features{%
2304     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
2305     \MT@font \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
2306     \ifMT@inlist@
2307       \MT@let@nc{\MT@nameuse{MT@abbr@##1}}\relax
2308     \else
2309       \MT@if@false
2310     \fi
2311   }%
2312   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2313 }

```

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

```

2314 \def\MT@register@subst@font@cx{%
2315   \MT@map@clist@c\MT@active@features{%
2316     \expandafter\MT@xadd
2317     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
2318     {\font@name,}%
2319   }%
2320 }

```

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

```

2321 \def\MT@register@font@cx{%
2322   \MT@map@clist@c\MT@active@features{%
2323     \expandafter\ifx\csname MT@nameuse{MT@abbr@##1}\endcsname\relax\else
2324     \expandafter\MT@xadd
2325     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
2326     {\MT@font,}%
2327     \def\@tempa{##1}%
2328     \expandafter\MT@map@tlist@c

```

```

2329      \csname MT@##1@doc@contexts\endcsname
2330      \MT@maybe@rem@from@list
2331      \fi
2332    }%
2333  }

```

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

```

2334 \def\MT@maybe@rem@from@list#1{%
2335   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
2336     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
2337     \MT@font \csname MT@\@tempa @#1font@list\endcsname
2338   }%
2339 }

```

`\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.

```

2340 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}
2341 \MT@addto@setup{%
2342   \DeclareRobustCommand\microtypecontext[1]{%
2343     \MT@setup@contexts
2344     \let\MT@reset@context\relax
2345     \setkeys{MTC}{#1}%
2346     \selectfont
2347     \MT@reset@context
2348   }%
2349 }

```

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

```

2350 \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@`

```

2351 \def\MT@reset@context@{%
2352   \MT@vinfo{<<< Resetting contexts\on@line
2353   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
2354   <debug> / \MT@tr@context/\MT@kn@context/\MT@sp@context
2355   }%
2356   \selectfont
2357 }

```

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

```

2358 \def\MT@setup@contexts{%
2359   \MT@map@clist@c\MT@active@features
2360   {\global\MT@let@nc{MT@##1@font@list}\MT@font@list}%
2361   \MT@gl@et\MT@check@font\MT@check@font@cx
2362   \MT@gl@et\MT@register@font\MT@register@font@cx
2363   \MT@gl@et\MT@register@subst@font\MT@register@subst@font@cx
2364   \MT@gl@et\MT@setup@contexts\relax
2365 }

```

Define context keys.

```

2366 \MT@map@clist@c\MT@features@long{%
2367   \define@key{MTC}{#1}[]{}%
2368   \edef\@tempb{\@nameuse{MT@rbb@#1}}%
2369   \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2370   \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).

```

2371 \MT@ifempty{##1}{\def\MT@val{0}}{\def\MT@val{##1}}%
2372 \expandafter\ifx\csname MT@\@tempb @context\endcsname\MT@val
2373 (debug) \MT@info{1}{>>> no change of #1 context: `\'MT@val'}%
2374 \else
2375 \MT@vinfo{>>> Changing #1 context to `\'MT@val'\MessageBreak\on@line
2376 (debug) \space(previous: `\'@nameuse{MT@\@tempb @context}')}%
2377 }%
2378 \def\MT@reset@context{\aftergroup\MT@reset@context@}%

```

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

```

2379 \global\MT@let@nn{\MT@reset@\@tempb @codes}{\MT@reset@\@tempb @codes}%

```

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

```

2380 \expandafter\MT@exp@one@\expandafter\MT@in@tlist\expandafter
2381 \MT@val \csname MT@\@tempb @doc@contexts\endcsname
2382 \ifMT@inlist@ \else
2383 \expandafter\MT@xadd\csname MT@\@tempb @doc@contexts\endcsname{{\MT@val}}%
2384 (debug) \MT@info{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
2385 \fi
2386 \MT@edef@n{\MT@\@tempb @context}{\MT@val}%
2387 \fi
2388 \fi
2389 }%
2390 }

```

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

`\MT@ex@context` 2391 `\MT@exp@one@\MT@map@clist@\MT@features,nl}{%`

`\MT@tr@context` 2392 `\MT@def@n{\MT@#1@context}{@}%`

`\MT@sp@context` 2393 `\MT@def@n{\MT@#1@doc@contexts}{{@}}%`

`\MT@kn@context` 2394 `}`

`\MT@kn@context` 2395 `\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.

```

2396 \def\DeclareMicrotypeSet{%
2397 \ifstar
2398 \MT@DeclareSetAndUseIt
2399 \MT@DeclareSet
2400 }
\MT@DeclareSet
2401 \newcommand\MT@DeclareSet[3][]{%
2402 \KV@sp@def\@tempa{#1}%
2403 \MT@ifempty\@tempa{%
2404 \MT@map@clist@c\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%
2405 }{%
2406 \MT@map@clist@c\@tempa{%
2407 \KV@sp@def\@tempa{##1}%

```

```

2408 \MT@ifempty\@tempa\relax{%
2409 \MT@is@feature{set declaration `#2'}{%
2410 \MT@exp@one@n\MT@declare@sets
2411 {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
2412 }%
2413 }%
2414 }}%
2415 }%
2416 }

```

\MT@DeclareSetAndUseIt

```

2417 \newcommand\MT@DeclareSetAndUseIt[3][[]]{%
2418 \MT@DeclareSet[#1]{#2}{#3}%
2419 \UseMicrotypeSet[#1]{#2}%
2420 }

```

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

```
2421 \let\MT@curr@set@name\@empty
```

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

```

2422 \def\MT@declare@sets#1#2#3{%
2423 \KV@sp@def\MT@curr@set@name{#2}%
2424 \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
2425 \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2426 \global\MT@let@nc{MT@#1list@size@\MT@curr@set@name}\@empty
2427 }%
2428 \global\MT@let@nc{MT@#1@set@@\MT@curr@set@name}\@empty
2429 (debug)\MT@edinfo{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2430 \setkeys{MT@#1@set}{#3}%
2431 }

```

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

```

2432 \def\MT@define@set@key@#1#2{%
2433 \define@key{MT@#2@set}{#1}[]{%
2434 \global\MT@let@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
2435 \MT@map@clist@n{##1}{%
2436 \KV@sp@def\MT@val{###1}%
2437 \MT@get@highlevel{#1}%

```

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

```

2438 \MT@exp@two@n\g@addto@macro
2439 {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
2440 {\MT@val,}%
2441 }%

```

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

```

2442 \expandafter\g@addto@macro\expandafter\MT@font@sets
2443 \csname MT@#2list@#1@\MT@curr@set@name\endcsname
2444 (debug)\MT@edinfo{n1}{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
2445 }%
2446 }

```

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

```

2447 \def\MT@get@highlevel#1{%
2448 \expandafter\MT@test@ast\MT@val*\@nil\relax{%

```

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

```

2449 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
2450 \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.

```
2451 }%
2452 }
```

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

```
2453 \def\MT@test@ast#1*#2\@nil{%
2454   \def\@tempa{#1}%
2455   \MT@ifempty{#2}%
2456 }
```

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

```
\MT@fix@font@set 2457 \let\MT@font@sets\empty
2458 \def\MT@fix@font@set#1{%
2459   \xdef#1{#1}%
2460   \global\@onelevel\@sanitize#1%
2461 }
```

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

```
2462 \def\MT@define@set@key@size#1{%
2463   \define@key{MT@#1@set}{size}[]{}%
2464   \MT@map@clist{\##1}{%
2465     \KV@esp@def\MT@val{###1}%
2466     \expandafter\MT@get@range\MT@val--\@nil
2467     \ifx\MT@val\relax \else
2468       \expandafter\MT@xadd
2469       \csname MT@#1list@size@\MT@curr@set@name\endcsname
2470       {{{\MT@lower}{\MT@upper}\relax}}%
2471     \fi
2472   }%
2473   <debug>\MT@info{\##1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
2474   }%
2475 }
```

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

`\MT@get@range` Ranges will be stored as triplets of `{\lower bound}{\upper bound}{\list name}`.

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

```
\MT@lower 2476 \def\MT@get@range#1-#2-#3\@nil{%
2477   \MT@ifempty{#1}{%
2478     \MT@ifempty{#2}{%
2479       \let\MT@val\relax
2480     }%
2481     \def\MT@lower{0}%
2482     \def\MT@val{#2}%
2483     \MT@get@size
2484     \edef\MT@upper{\MT@val}%
2485   }%
2486 }%
2487 \def\MT@val{#1}%
2488 \MT@get@size
2489 \ifx\MT@val\relax \else
2490   \edef\MT@lower{\MT@val}%
2491   \MT@ifempty{#2}{%
2492     \MT@ifempty{#3}%
2493   }
```

```

2493      {\def\MT@upper{-1}}%
2494      2048pt is TEX's maximum font size.
2495      {\def\MT@upper{2048}}%
2496      }{%
2497      \def\MT@val{#2}%
2498      \MT@get@size
2499      \ifx\MT@val\relax \else
2500      \MT@ifdim\MT@lower>\MT@val{%
2501      \MT@error{%
2502      Invalid size range (\MT@lower\space > \MT@val) in font set
2503      ~\MT@curr@set@name'.\MessageBreak Swapping sizes}}%
2504      \edef\MT@upper{\MT@lower}%
2505      \edef\MT@lower{\MT@val}%
2506      }{%
2507      \edef\MT@upper{\MT@val}%
2508      }%
2509      \MT@ifdim\MT@lower=\MT@upper
2510      {\def\MT@upper{-1}}%
2511      \relax
2512    \fi
2513  }%
2514 }
2515 }

```

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

```

2516 \def\MT@get@size{%
2517   A single star would mean \sizedefault, which doesn't exist, so we define it to be
2518   \normalsize.
2519   \if*\MT@val\relax
2520     \def\@tempa{\normalsize}%
2521   \else
2522     \MT@let@cn\@tempa{\MT@val}%
2523   \fi
2524   \ifx\@tempa\relax \else

```

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

```

2523   \begingroup
2524     \def\set@fontsize##1##2##3##4\@nil{\gdef\MT@val{##2}}%
2525     \@tempa\@nil
2526   \endgroup
2527   \fi

```

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

```

2528   \MT@ifdimen\MT@val{%
2529     \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2530     \edef\MT@val{\strip@pt\@tempdima}%
2531   }{%
2532     \MT@warning{Could not parse font size ~\MT@val'\MessageBreak
2533     in font set ~\MT@curr@set@name'}%
2534     \let\MT@val\relax
2535   }%
2536 }

```

\MT@define@set@key@font

```

2537 \def\MT@define@set@key@font#1{%
2538   \define@key{MT@#1@set}{font}[]{}%
2539   \global\MT@let\nc{MT@#1list@font@MT@curr@set@name}\@empty
2540   \MT@map@clist@n{##1}{%
2541     \KV@esp@def\MT@val{###1}%
2542     \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/}}\relax
2543     \expandafter\MT@get@font\MT@val////\@nil
2544     \MT@exp@two@n@g@addto@macro
2545       {\csname MT@#1list@font@MT@curr@set@name\expandafter\endcsname}%
2546       {\MT@val,}%
2547   }%
2548   \expandafter\g@addto@macro\expandafter\MT@font@sets
2549     \csname MT@#1list@font@MT@curr@set@name\endcsname
2550   <debug>\MT@edinfo@n1{1}{-- font: \@nameuse{MT@#1list@font@MT@curr@set@name}}%
2551   }%
2552 }

```

\MT@get@font Translate any asterisks.

```

2553 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2554   \MT@ifempty{#1#2#3#4#5}\relax{%
2555     \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
2556     \ifx\MT@val\relax\def\MT@val{0}\fi
2557     \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2558     \let\MT@val\@tempb
2559   }%
2560 }

```

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

```

2561 \def\MT@get@font@#1#2#3#4#5#6{%
2562   \let\@tempb\@empty
2563   \def\MT@temp{#1/#2/#3/#4/#5}%
2564   \MT@get@axis{encoding}{#1}%
2565   \MT@get@axis{family}{#2}%
2566   \MT@get@axis{series}{#3}%
2567   \MT@get@axis{shape}{#4}%
2568   \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
2569   \MT@ifempty{#5}{%
2570     \MT@warn@axis@empty{size}{\string\normalsize}%
2571     \def\MT@val{*}%
2572   }{%
2573     \def\MT@val{#5}%
2574   }%
2575   \MT@get@size
2576 }

```

\MT@get@axis

```

2577 \def\MT@get@axis#1#2{%
2578   \def\MT@val{#2}%
2579   \MT@get@highlevel{#1}%
2580   \MT@ifempty\MT@val{%
2581     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2582     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2583   }\relax
2584   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2585 }

```

\MT@warn@axis@empty

```

2586 \def\MT@warn@axis@empty#1#2{%
2587   \MT@warning{#1 axis is empty in font specification\MessageBreak
2588     ~\MT@temp'. Using ~#2' instead}%
2589 }

```

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

```
2590 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
2591   \MT@define@set@key@{encoding}{#1}%
2592   \MT@define@set@key@{family}{#1}%
2593   \MT@define@set@key@{series}{#1}%
2594   \MT@define@set@key@{shape}{#1}%
2595   \MT@define@set@key@size{#1}%
2596   \MT@define@set@key@font{#1}%
2597 }
```

`\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.

```
2598 \renewcommand*\UseMicrotypeSet[2][]{%
2599   \KV@sp@def\@tempa{#1}%
2600   \MT@ifempty\@tempa{%
2601     \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
2602   }{%
2603     \MT@map@clist@c\@tempa{%
2604       \KV@sp@def\@tempa{##1}%
2605       \MT@ifempty\@tempa\relax{%
2606         \MT@is@feature{activation of set `#2'}{%
2607           \MT@exp@one@n\MT@use@set
2608             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2609         }%
2610       }%
2611     }%
2612   }%
2613 }
```

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

```
\MT@ex@setname 2614 \def\MT@use@set#1#2{%
\MT@tr@setname 2615   \KV@sp@def\@tempa{#2}%
2616   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@sp@setname 2617     \MT@xdef@n{MT@#1@setname}{\@tempa}%
\MT@kn@setname 2618   }{%
2619     \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2620       \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2621     }%
2622     \MT@error{%
2623       The \@nameuse{MT@abbr@#1} set `@\@tempa' is undeclared.\MessageBreak
2624       Using set `@\@nameuse{MT@#1@setname}' instead}{}%
2625   }%
2626 }
```

`\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.

```
2627 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2628   \KV@sp@def\@tempa{#1}%
2629   \MT@ifempty\@tempa{%
2630     \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
2631   }{%
2632     \MT@map@clist@c\@tempa{%
2633       \KV@sp@def\@tempa{##1}%
2634       \MT@ifempty\@tempa\relax{%
2635         \MT@is@feature{declaration of default set `#2'}{%
2636           \MT@exp@one@n\MT@set@default@set
2637             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2638         }%
2639       }%
2640     }%
2641   }
```



```

2641 }%
2642 }

\MT@default@pr@set
\MT@default@ex@set 2643 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 2644 \KV@sp@def\@tempa{#2}%
2645 \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@default@sp@set 2646 (debug)\MT@info{1}{declaring default \@nameuse{MT@abbr@#1} set \@tempa'}%
\MT@default@kn@set 2647 \MT@xdef@n{MT@default@#1@set}{\@tempa}%
\MT@set@default@set 2648 }{%
2649 \MT@error{%
2650 The \@nameuse{MT@abbr@#1} set \@tempa' is not declared.\MessageBreak
2651 Cannot make it the default set. Using set\MessageBreak `all' instead}{}%
2652 \MT@xdef@n{MT@default@#1@set}{all}%
2653 }%
2654 }

```

### 14.3.2 Variants and Aliases

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

```

2655 \let\MT@variants\@empty
2656 \def\DeclareMicrotypeVariants{%
2657 \ifstar
2658 \MT@DeclareVariants
2659 {\let\MT@variants\@empty\MT@DeclareVariants}%
2660 }

```

`\MT@DeclareVariants`

```

2661 \def\MT@DeclareVariants#1{%
2662 \MT@map@clist@n{#1}{%
2663 \KV@sp@def\@tempa{##1}%
2664 \@onelevel@sanitize\@tempa
2665 \xdef\MT@variants{\MT@variants{\@tempa}}%
2666 }%
2667 }

```

`\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.

```

2668 \renewcommand*\DeclareMicrotypeAlias[2]{%
2669 \KV@sp@def\@tempa{#1}%
2670 \KV@sp@def\@tempb{#2}%
2671 \@onelevel@sanitize\@tempb
2672 \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2673 \MT@warning{Alias font family \@tempb' will override
2674 alias \@nameuse{MT@\@tempa @alias}'\MessageBreak
2675 for font family \@tempa'}}%
2676 \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.

```

2677 \MT@ifdefined@c@T\MT@family{%
2678 (debug)\MT@info{1}{Activating alias font \@tempb' for \@MT@family'}%
2679 \MT@glet\MT@familyalias\@tempb
2680 }%
2681 }

```

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

```

2682 \def\LoadMicrotypeFile#1{%

```

```

2683 \KV@sp@def\@tempa{#1}%
2684 \@onelevel@sanitize\@tempa
2685 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2686 \ifMT@inlist@
2687 \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2688 \else
2689 \MT@xadd\MT@file@list{\@tempa,}%
2690 \MT@begin@catcodes
2691 \InputIfFileExists{mt-\@tempa.cfg}{%
2692 \edef\MT@curr@file{mt-\@tempa.cfg}%
2693 \MT@vinfo{... Loading configuration file \MT@curr@file}%
2694 }{%
2695 \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2696 does not exist}%
2697 }%
2698 \MT@end@catcodes
2699 \fi
2700 }

```

### 14.3.3 Disabling Ligatures

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

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

```

\MT@nl@ligatures 2701 \MT@requires@pdftex5{
2702 \def\DisableLigatures{%
2703 \MT@begin@catcodes
2704 \MT@DisableLigatures
2705 }
2706 \newcommand*\MT@DisableLigatures[2][{}]{%
2707 \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
2708 \xdef\MT@active@features{\MT@active@features,nl}%
2709 \global\MT@noligaturestrue
2710 \MT@declare@sets{nl}{no ligatures}{#2}%
2711 \gdef\MT@nl@setname{no ligatures}%
2712 \MT@end@catcodes
2713 }
2714 }{
2715 \renewcommand*\DisableLigatures[2][{}]{%
2716 \MT@error{Disabling ligatures of a font is only possible\MessageBreak
2717 with pdftex version 1.30 or newer.\MessageBreak
2718 Ignoring \string\DisableLigatures}{Upgrade pdftex.}%
2719 }
2720 }

```

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

### 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.

```

2721 \def\DeclareMicrotypeBabelHook#1#2{%
2722 \MT@map@clist@n{#1}{%
2723 \KV@sp@def\@tempa{##1}%
2724 \MT@gdef@n{\MT@babel@{\@tempa}{#2}}%
2725 }%
2726 }

```

### 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).

```
2727 \def\SetProtrusion{%
2728   \MT@begin@catcodes
2729   \MT@SetProtrusion
2730 }
```

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

```
\MT@pr@c@name 2731 \newcommand*\MT@SetProtrusion[3] [] {%
```

```
\MT@extra@context 2732   \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.

```
2733   \MT@set@named@keys{MT@pr@c}{#1}%
2734   <debug>\MT@info{1}{creating protrusion list `~\MT@pr@c@name'}%
2735   \def\MT@permutelist{pr@c}%
2736   \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>`, ...

```
2737   \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.

```
2738   \MT@gdefn{MT@pr@c@MT@pr@c@name}{#3}%
2739   \MT@end@catcodes
2740 }
```

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

```
2741 \def\SetExpansion{%
2742   \MT@begin@catcodes
2743   \MT@SetExpansion
2744 }
```

`\MT@SetExpansion`

```
\MT@ex@c@name 2745 \newcommand*\MT@SetExpansion[3] [] {%
```

```
\MT@extra@context 2746   \let\MT@extra@context\@empty
```

```
\MT@permutelist 2747   \MT@set@named@keys{MT@ex@c}{#1}%
```

```
2748   \MT@ifdefinedn@T{MT@ex@c@MT@ex@c@name @factor}{%
2749     \ifnum\csname MT@ex@c@MT@ex@c@name @factor\endcsname > \@m
2750       \MT@warningn1{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
2751         too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
2752         maximum of 1000}%
2753     \global\MT@let@nc{MT@ex@c@MT@ex@c@name @factor}\@m
2754   \fi
2755 }
```

```
2756   <debug>\MT@info{1}{creating expansion list `~\MT@ex@c@name'}%
```

```
2757   \def\MT@permutelist{ex@c}%
```

```
2758   \setkeys{MT@cfg}{#2}%
```

```
2759   \MT@permute
```

```
2760   \MT@gdefn{MT@ex@c@MT@ex@c@name}{#3}%
```

```
2761   \MT@end@catcodes
```

```

2762 }

\SetTracking
2763 \def\SetTracking{%
2764   \MT@begin@catcodes
2765   \MT@SetTracking
2766 }

\MT@SetTracking    Third argument may be empty.
2767 \newcommand*\MT@SetTracking[3] [] {%
2768   \let\MT@extra@context\@empty
2769   \MT@set@named@keys{MT@tr@{}}{#1}%
2770   (debug)\MT@dinfor{1}{creating tracking list `\'MT@tr@{c@name}'}%
2771   \def\MT@permutelist{tr@c}%
2772   \setkeys{MT@cfg}{#2}%
2773   \MT@permute
2774   \KV@sp@def\@tempa{#3}%
2775   \MT@ifempty\@tempa\relax{%
2776     \MT@ifint\@tempa
2777     {\MT@xdef\@n{MT@tr@c@\'MT@tr@c@name}{\'@tempa}}%
2778     {\MT@warning{Value `\'@tempa' is not a number in\MessageBreak
2779       tracking set `\'MT@curr@set@name'}}}%
2780   \MT@end@catcodes
2781 }

\SetExtraSpacing
2782 \def\SetExtraSpacing{%
2783   \MT@begin@catcodes
2784   \MT@SetExtraSpacing
2785 }

\MT@SetExtraSpacing
\MT@sp@c@name 2786 \newcommand*\MT@SetExtraSpacing[3] [] {%
\MT@extra@context 2787   \let\MT@extra@context\@empty
2788   \MT@set@named@keys{MT@sp@c@{}}{#1}%
\MT@permutelist 2789   (debug)\MT@dinfor{1}{creating spacing list `\'MT@sp@c@name}'}%
2790   \def\MT@permutelist{sp@c}%
2791   \setkeys{MT@cfg}{#2}%
2792   \MT@permute
2793   \MT@gdef\@n{MT@sp@c@\'MT@sp@c@name}{#3}%
2794   \MT@end@catcodes
2795 }

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

\MT@SetExtraKerning
\MT@kn@c@name 2800 \newcommand*\MT@SetExtraKerning[3] [] {%
\MT@extra@context 2801   \let\MT@extra@context\@empty
2802   \MT@set@named@keys{MT@kn@c@{}}{#1}%
\MT@permutelist 2803   (debug)\MT@dinfor{1}{creating kerning list `\'MT@kn@c@name}'}%
2804   \def\MT@permutelist{kn@c}%
2805   \setkeys{MT@cfg}{#2}%
2806   \MT@permute
2807   \MT@gdef\@n{MT@kn@c@\'MT@kn@c@name}{#3}%
2808   \MT@end@catcodes
2809 }

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

```

remaining keys.

```

2810 \def\MT@set@named@keys#1#2{%
2811   \def\x##1name=##2,##3\@nil{%
2812     \setkeys{#1}{name=##2}%
2813     \def\MT@options{##1##3}%
2814     \MT@rem@from@clist{name=}\MT@options
2815   }%
2816   \x#2,name=,\@nil
2817   \@expandtwoargs\setkeys{#1}\MT@options
2818 }

```

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

```

2819 \def\MT@define@code@key#1#2{%
2820   \define@key{MT@#2}{#1}[]{%
2821     \@tempcnta=\@ne
2822     \MT@map@cliston{##1}{%
2823       \KV@sp@def\MT@val{###1}%

```

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

```

2824     \MT@get@highlevel{#1}%
2825     \MT@edefon{MT@temp#1\the\@tempcnta}{\MT@val}%
2826     \advance\@tempcnta \@ne
2827   }%
2828 }%
2829 }

```

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

```

2830 \def\MT@define@code@key@size#1{%
2831   \define@key{MT@#1}{size}[]{%
2832     \MT@map@cliston{##1}{%
2833       \KV@sp@def\MT@val{###1}%
2834       \expandafter\MT@get@range\MT@val--\@nil
2835       \ifx\MT@val\relax \else
2836         \expandafter\MT@xadd\csname MT@tempsize\endcsname
2837         {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2838       \fi
2839     }%
2840   }%
2841 }

```

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

```

2842 \def\MT@define@code@key@font#1{%
2843   \define@key{MT@#1}{font}[]{%
2844     \MT@map@cliston{##1}{%
2845       \KV@sp@def\MT@val{###1}%
2846       \MT@ifstreq\MT@val*{\def\MT@val{*/**/*/*}}\relax
2847       \expandafter\MT@get@font@and@size\MT@val////\@nil
2848       \MT@xdefon{MT@MT@permutelist @\@tempb\MT@extra@context}%
2849       {\csname MT@MT@permutelist @name\endcsname}%
2850       (debug) \MT@edinfoenl{1}{initialising: use list for font \@tempb=\MT@val
2851       (debug) \ifx\MT@extra@context\@empty\else\MessageBreak
2852       (debug) (context: \MT@extra@context)\fi}%
2853       \expandafter\MT@xaddb
2854       \csname MT@MT@permutelist @\@tempb\MT@extra@context @sizes\endcsname
2855       {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
2856     }%
2857   }%
2858 }

```

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

```

2859 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%

```

```

2860 \MT@ifempty{#1#2#3#4#5}\relax
2861 {\MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}}%
2862 }

2863 \MT@define@code@key{encoding}{cfg}
2864 \MT@define@code@key{family}{cfg}
2865 \MT@define@code@key{series}{cfg}
2866 \MT@define@code@key{shape}{cfg}
2867 \MT@define@code@key@size{cfg}
2868 \MT@define@code@key@font{cfg}

```

\MT@define@opt@key

```

2869 \def\MT@define@opt@key#1#2{%
2870 \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
2871 \MT@xdef@n{MT@#1@c@MT@curr@set@name @#2}{##1}}}%
2872 }

```

The options in the optional first argument.

```
2873 \MT@map@clist@{MT@features}{%
```

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

```

2874 \define@key{MT@#1@c}{name}[]{%
2875 \MT@ifempty{##1}{%
2876 \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2877 }{%
2878 \MT@edef@n{MT@#1@c@name}{##1}%
2879 \MT@ifdefined@n@T{MT@#1@c@csname MT@#1@c@name\endcsname}{%
2880 \MT@warning{Redefining \nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}'}%
2881 }%
2882 }%
2883 \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
2884 }%
2885 \MT@define@opt@key{#1}{load}%
2886 \MT@define@opt@key{#1}{factor}%
2887 \MT@define@opt@key{#1}{preset}%
2888 \MT@define@opt@key{#1}{inputenc}%

```

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

```

2889 \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
2890 }

```

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

```

2891 \MT@requires@pdftex7{
2892 \lua
2893 \MT@requires@luatex{
2894 \define@key{MT@ex@c}{context}[]{%
2895 \MT@error{Expansion contexts currently don't work with luatex.\MessageBreak
2896 Ignoring `context' key\on@line}%
2897 {Use pdftex instead.}%
2898 }
2899 }{
2900 \lua
2901 \define@key{MT@ex@c}{context}[]{\MT@ifempty{##1}\relax{%
2902 \MT@gl@et\MT@copy@font\MT@copy@font@
2903 \def\MT@extra@context{##1}}}%
2904 }
2905 \MT@addto@setup{%
2906 \define@key{MT@ex@c}{context}[]{%
2907 \ifx\MT@copy@font\MT@copy@font@

```

```

2908     \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
2909     \else
2910     \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
2911         Ignoring `context' key\on@line}%
2912     {Either move the settings inside the preamble,\MessageBreak
2913     or load the package with the `copyfonts' option.}%
2914     \fi
2915 }
2916 }
2917 \Lua }
2918 }{
2919 \define@key{MT@ex@c}{context}[]{%
2920 \MT@error{Expansion contexts only work with pdfTeX 1.40.4\MessageBreak
2921 or later. Ignoring `context' key\on@line}%
2922 {Upgrade pdfTeX.}%
2923 }
2924 }

```

\MT@warn@nodim

```

2925 \def\MT@warn@nodim#1{%
2926 \MT@warning{`@tempa' is not a dimension.\MessageBreak
2927 Ignoring it and setting values relative to\MessageBreak #1}%
2928 }

```

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

```

2929 \define@key{MT@pr@c}{unit}[character]{%
2930 \global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
2931 \def\@tempa{#1}%
2932 \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.

```

2933 \MT@ifdimen\@tempa
2934 {\global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
2935 {\MT@warn@nodim{character widths}}%
2936 }%
2937 }

```

Tracking may only be relative to a dimension.

```

2938 \define@key{MT@tr@c}{unit}[1em]{%
2939 \global\MT@let@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
2940 \def\@tempa{#1}%
2941 \MT@ifdimen\@tempa
2942 {\global\MT@let@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
2943 {\MT@warn@nodim{1em}}%
2944 \MT@gdefn{MT@tr@c@MT@curr@set@name @unit}{1em}}%
2945 }

```

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

```

2946 \MT@map@clist@n{sp,kn}{%
2947 \define@key{MT@#1@c}{unit}[space]{%
2948 \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
2949 \def\@tempa{##1}%
2950 \MT@ifstreq\@tempa{character}\relax{%
2951 \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
2952 \MT@ifstreq\@tempa{space}\relax{%
2953 \MT@ifdimen\@tempa
2954 {\global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
2955 {\MT@warn@nodim{width of space}}%
2956 }%
2957 }%
2958 }%

```

2959 }

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

```
2960 \MT@map@clist@n{stretch,shrink,step}{%
2961   \define@key{MT@ex@c}{#1}[]{}%
2962   \MT@ifempty{##1}\relax{%
2963     \MT@ifint{##1}{%
```

A space terminates the number.

```
2964     \MT@gdef@n{MT@ex@c@MT@curr@set@name @#1}{##1 }%
2965   }{}%
2966   \MT@warning{%
2967     Value `##1' for option `~1' is not a number.\MessageBreak
2968     Ignoring it}%
2969   }%
2970 }%
2971 }%
2972 }
2973 \define@key{MT@ex@c}{auto}[true]{%
2974   \def\@tempa{#1}%
2975   \csname if\@tempa\endcsname
```

Don't use `autoexpand` for pdfTeX version older than 1.20.

```
2976   \MT@requires@pdfTeX4{%
2977     \MT@gdef@n{MT@ex@c@MT@curr@set@name @auto}{autoexpand}%
2978   }{}%
2979   \MT@warning{pdfTeX too old for automatic font expansion}%
2980 }
2981 \else
2982   \MT@requires@pdfTeX4{%
2983     \global\MT@let@nc{MT@ex@c@MT@curr@set@name @auto}\@empty
2984   }\relax
2985 \fi
2986 }
```

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

```
2987 \MT@define@opt@key{tr}{spacing}
2988 \MT@define@opt@key{tr}{outerspacing}
2989 \MT@define@opt@key{tr}{outerkerning}
```

Which ligatures should be disabled?

```
2990 \define@key{MT@tr@c}{noligatures}[]{}%
2991   {\MT@xdef@n{MT@tr@c@MT@curr@set@name @noligatures}{#1}}
2992 \define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
2993 \define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
2994 \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, and to specify an input encoding.

`\MT@extra@inputenc`



```

2995 \renewcommand*\DeclareCharacterInheritance[1] [] {%
2996   \let\MT@extra@context\@empty
2997   \let\MT@extra@inputenc\@undefined
2998   \let\MT@inh@feat\@empty
2999   \setkeys{MT@inh@}{#1}%
3000   \MT@begin@catcodes
3001   \MT@set@inh@list
3002 }

```

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

```

3003 \def\MT@set@inh@list#1#2{%
3004   \MT@ifempty\MT@inh@feat{%
3005     \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
3006   }%
3007   \MT@map@clist@c\MT@inh@feat{{%
3008     \KV@esp@def\@tempa{##1}%
3009     \MT@ifempty\@tempa\relax{%
3010       \MT@exp@one@n\MT@declare@char@inh
3011       {csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
3012     }%
3013   }}%
3014 }%
3015 \MT@end@catcodes
3016 }

```

The keys for the optional argument.

```

3017 \MT@map@clist@c\MT@features@long{%
3018   \define@key{MT@inh@}{#1} [] {\edef\MT@inh@feat{\MT@inh@feat#1,}}%
3019   \define@key{MT@inh@}{inputenc} {\def\MT@extra@inputenc{#1}}

```

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

```

3020 \def\MT@declare@char@inh#1#2#3{%
3021   \MT@edef@n{MT@#1@inh@name}%
3022   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3023   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3024   \MT@ifdefined@c@T\MT@extra@inputenc{%
3025     \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3026   (debug) \MT@info{1}{creating inheritance list ~\@nameuse{MT@#1@inh@name}}%
3027   \MT@gdef@n{MT@#1@inh@csname MT@#1@inh@name\endcsname}{#3}%
3028   \def\MT@permutelist{#1@inh}%
3029   \setkeys{MT@inh}{#2}%
3030   \MT@permute
3031 }

```

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

```

3032 \define@key{MT@inh}{encoding} [] {%
3033   \def\MT@val{#1}%
3034   \expandafter\MT@encoding@check\MT@val,\@nil
3035   \MT@get@highlevel{encoding}%
3036   \MT@edef@n{MT@tempencoding1}{\MT@val}%
3037 }

```

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

```

3038 \def\MT@encoding@check#1,#2\@nil{%
3039   \MT@ifempty{#2}\relax{%
3040     \edef\MT@val{#1}%
3041     \MT@warning{You may only specify one encoding for character\MessageBreak
3042               inheritance lists. Ignoring encoding(s) #2}%
3043   }%
3044 }

```

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

```

3045 \MT@define@code@key{family}{inh}
3046 \MT@define@code@key{series}{inh}
3047 \MT@define@code@key{shape}{inh}
3048 \MT@define@code@key{size}{inh}
3049 \MT@define@code@key{font}{inh}

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

3050 \def\MT@inh@do#1,{%
3051   \ifx\relax#1\@empty \else
3052     \MT@inh@split #1==\relax
3053     \expandafter\MT@inh@do
3054   \fi
3055 }

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

3056 \def\MT@inh@split#1=#2=#3\relax{%
3057   \def\@tempa{#1}%
3058   \ifx\@tempa\@empty \else
3059     \MT@get@slot
3060     \ifnum\MT@char > \m@ne
3061       \let\MT@val\MT@char
3062       \MT@map@clist@n{#2}{%
3063         \def\@tempa{##1}%
3064         \ifx\@tempa\@empty \else
3065           \MT@get@slot
3066           \ifnum\MT@char > \m@ne
3067             \expandafter\MT@xadd
3068             \csname MT@inh@\MT@listname @\MT@val @\endcsname
3069             {\MT@char}}%
3070         \fi
3071       \fi
3072   }%
3073   <debug>\MT@info@n1{2}{children of #1 (\MT@val):
3074   <debug>         \@nameuse{MT@inh@\MT@listname @\MT@val @}}%
3075   \fi
3076 \fi
3077 }
```

### 14.3.7 Permutation

`\MT@permute` Calling `\MT@permute` will define commands for all permutations of the specified font attributes of the form `\MT@<list type>@<encoding>/<family>/<series>/<shape>/<|*>` to be the expansion of `\MT@<list type>@name`, i. e., the name of the currently defined list. Size ranges are held in a separate macro called `\MT@<list type>@<font axes>@sizes`, which in turn contains the respective *<list name>*s attached to the ranges.

```

3078 \def\MT@permute{%
3079   \let\MT@cnt@encoding\@ne
3080   \MT@permute@
```

Undefine commands for the next round.

```

3081 \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
3082 \MT@glet\MT@tempsize\@undefined
```

```

3083 }
3084 \def\MT@permute@{%
3085   \let\MT@cnt@family\@ne
3086   \MT@permute@@
3087   \MT@increment\MT@cnt@encoding
3088   \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
3089   \MT@permute@
3090 }
3091 \def\MT@permute@@{%
3092   \let\MT@cnt@series\@ne
3093   \MT@permute@@@
3094   \MT@increment\MT@cnt@family
3095   \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
3096   \MT@permute@@
3097 }
3098 \def\MT@permute@@@{%
3099   \let\MT@cnt@shape\@ne
3100   \MT@permute@@@@
3101   \MT@increment\MT@cnt@series
3102   \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
3103   \MT@permute@@@
3104 }
3105 \def\MT@permute@@@@{%
3106   \MT@permute@@@@@
3107   \MT@increment\MT@cnt@shape
3108   \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
3109   \MT@permute@@@@
3110 }

```

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

```

3111 \def\MT@permute@@@@@{%
3112   \MT@permute@define{encoding}%
3113   \ifMT@document
3114     \ifx\MT@tempencoding\empty \else
3115       \MT@ifdefined@n@TF{T@MT@tempencoding}\relax
3116       {\expandafter\expandafter\expandafter\@gobble}%
3117     \fi
3118   \fi
3119   \MT@permute@@@@@
3120 }

```

\MT@permute@@@@@

```

3121 \def\MT@permute@@@@@{%
3122   \MT@permute@define{family}%
3123   \MT@permute@define{series}%
3124   \MT@permute@define{shape}%
3125   \edef\@tempa{\MT@tempencoding
3126     /\MT@tempfamily
3127     /\MT@tempseries
3128     /\MT@tempshape
3129     /\MT@ifdefined@c@T\MT@tempsize *}

```

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

```

3130 \def\@tempb{////}%
3131 \ifx\@tempa\@tempb \else
3132   \ifx\MT@tempencoding\empty
3133     \MT@warning{%
3134       You have to specify an encoding for\MessageBreak
3135       \@nameuse{MT@abbr@MT@permute!list} list
3136       ~\@nameuse{MT@MT@permute!list @name}'.\MessageBreak
3137       Ignoring it}%

```

```

3138 \else
3139 \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

3140 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3141 \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3142 }%
3143 \expandafter\MT@xaddb
3144 \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
3145 \MT@tempsize
3146 <debug>\MT@edinfo@n1{1}{initialising: use list for font \@tempa,\MessageBreak
3147 <debug> sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3148 <debug> @sizes\endcsname}%
3149 }{%

```

Only one list can apply to a given combination.

```

3150 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3151 \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3152 \@nameuse{MT@MT@permutelist @name}' will override list\MessageBreak
3153 \@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3154 for font \@tempa'}%
3155 }%
3156 <debug>\MT@edinfo@n1{1}{initialising: use list for font \@tempa
3157 <debug> \ifx\MT@extra@context\@empty\else\MessageBreak
3158 <debug> (context: \MT@extra@context)\fi}%
3159 }%
3160 \MT@xdef@n{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3161 { \csname MT@MT@permutelist @name\endcsname}%
3162 \fi
3163 \fi
3164 }

```

\MT@permute@define Define the commands.

```

3165 \def\MT@permute@define#1{%
3166 \expandafter\@tempcnta=\csname MT@cnt@#1\endcsname\relax
3167 \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3168 {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3169 {\MT@let@nc{MT@temp#1}\@empty}%
3170 }

```

\MT@permute@reset Reset the commands.

```

3171 \def\MT@permute@reset#1{%
3172 \@tempcnta=\@ne
3173 \MT@loop
3174 \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3175 \advance\@tempcnta\@ne
3176 \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3177 \iftrue
3178 \iffalse
3179 \MT@repeat
3180 }

```

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

```

3181 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

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

```

3182 \def\MT@check@rlist@#1#2#3{%
3183 \def\@tempb{#1}%
3184 \def\@tempc{#2}%
3185 \MT@iffalse

```

```

3186 \expandafter\MT@map@tlist@c
3187 \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
3188 \MT@check@range
3189 }

```

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

```

3190 \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.

```

3191 \def\MT@check@range@#1#2#3{%
3192   \MT@ifdim{#2}=\m@ne{%
3193     \MT@ifdim\@tempc=\m@ne{%

```

- Both items are simple sizes.

```

3194   \MT@ifdim\@tempb={#1}\MT@iftrue\relax
3195   }{%

```

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

```

3196   \MT@ifdim\@tempb>{#1}\relax{%
3197     \MT@ifdim\@tempc>{#1}{%
3198       \MT@iftrue
3199       \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3200     }\relax
3201   }%
3202 }%
3203 }{%
3204   \MT@ifdim\@tempc=\m@ne{%

```

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

```

3205   \MT@ifdim\@tempb<{#2}{%
3206     \MT@ifdim\@tempb<{#1}\relax\MT@iftrue
3207   }\relax
3208 }{%

```

- Both items are ranges.

```

3209   \MT@ifdim\@tempb<{#2}{%
3210     \MT@ifdim\@tempc>{#1}{%
3211       \MT@iftrue
3212       \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3213     }\relax
3214   }\relax
3215 }%
3216 }%
3217 \ifMT@if@
3218 \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
3219   \@nameuse{MT@\MT@permutelist @name}' will override\MessageBreak
3220   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.

```

3221 \expandafter\MT@tlist@break
3222 \fi
3223 }

```

## 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 3224 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 3225 \newif\ifMT@opt@auto
3226 \newif\ifMT@opt@DVI

\MT@optwarn@admissible   Some warnings.
3227 \def\MT@optwarn@admissible#1#2{%
3228   \MT@warning@n1{`#1' is not an admissible value for option\MessageBreak
3229     `#2'. Assuming `false'}%
3230 }

\MT@optwarn@nan
3231 </package>
3232 <plain>\ifnum\MT@plain<2
3233 \def\MT@optwarn@nan#1#2{%
3234   \MT@warning@n1{Value `#1' for option `#2' is not a\MessageBreak number.
3235     Using default value of \number\@nameuse{MT@#2@default}}%
3236 }
3237 <plain>\fi
3238 <*package>

\MT@opt@def@set
3239 \def\MT@opt@def@set#1{%
3240   \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}}{%
3241     \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
3242   }{%
3243     \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
3244     \MT@warning@n1{The #1 set `MT@val' is undeclared.\MessageBreak
3245       Using set ` \@nameuse{MT@\@tempb @setname}' instead}%
3246   }%
3247 }

expansion and protrusion may be true, false, compatibility, nocompatibility
and/or a <set name>.
3248 \MT@map@clist@n{protrusion,expansion}{%
3249   \define@key{MT}{#1}[true]{%
3250     \csname MT@opt@#1true\endcsname
3251     \MT@map@clist@n{##1}{%
3252       \KV@esp@def\MT@val{###1}%
3253       \MT@ifempty\MT@val\relax{%
3254         \csname MT@#1true\endcsname
3255         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3256         \MT@ifstreq\MT@val{true}\relax
3257       }%
3258       \MT@ifstreq\MT@val{false}{%
3259         \csname MT@#1false\endcsname
3260       }%
3261       \MT@ifstreq\MT@val{compatibility}{%
3262         \MT@let@nc{MT@\@tempb @level}\@ne
3263       }{%
3264         \MT@ifstreq\MT@val{nocompatibility}{%
3265           \MT@let@nc{MT@\@tempb @level}\tw@
3266         }%
3267       }%
3268     }%

```

If everything failed, it should be a set name.

```

3269         }%
3270     }%
3271 }%
3272 }%
3273 }%
3274 }%
3275 }

```

activate is a shortcut for protrusion and expansion.

```

3276 \define@key{MT}{activate}[true]{%
3277     \setkeys{MT}{protrusion={#1}}%
3278     \setkeys{MT}{expansion={#1}}%
3279 }

```

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

```

3280 \MT@map@clist@n{spacing,kerning,tracking}{%
3281     \define@key{MT}{#1}[true]{%
3282         \MT@map@clist@n{##1}{%
3283             \KV@sp@def\MT@val{###1}%
3284             \MT@ifempty\MT@val\relax{%
3285                 \csname MT@#1true\endcsname
3286                 \MT@ifstreq\MT@val{true}\relax
3287             }%
3288             \MT@ifstreq\MT@val{false}{%
3289                 \csname MT@#1false\endcsname
3290             }%
3291             \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3292             \MT@opt@def@set{#1}%
3293         }%
3294     }%
3295 }%
3296 }%
3297 }%
3298 }

```

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

```

3299 \def\MT@def@bool@opt#1#2{%
3300     \define@key{MT}{#1}[true]{%
3301         \def\@tempa{#1}%
3302         \MT@ifstreq\@tempa{true}\relax{%
3303             \MT@ifstreq\@tempa{false}\relax{%
3304                 \MT@optwarn@admissible{##1}{#1}%
3305                 \def\@tempa{false}%
3306             }%
3307         }%
3308         #2%
3309     }%
3310 }

```

Boolean options that only set the switch.

```

3311 \MT@map@clist@n{draft,selected,babel}{%
3312     \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
3313 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autottrue}

```

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

```

3314 \MT@def@bool@opt{DVIoutput}{%
3315     \csname if\@tempa\endcsname
3316     \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
3317     \pdfoutput\z@
3318 }

```

```

3319 \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
3320 \pdfoutput\@ne
3321 \fi
3322 }

```

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.

```

3323 \MT@def@bool@opt{defersetup}{%
3324 \csname if\@tempa\endcsname \else
3325 \AtEndOfPackage{%
3326 \MT@setup@
3327 \let\MT@setup@\empty
3328 \let\MT@addto@setup\@firstofone
3329 }%
3330 \fi
3331 }

```

`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.

```

3332 \MT@requires@pdftex7{
3333 (lua)
3334 \MT@requires@luatex{
3335 \MT@def@bool@opt{copyfonts}{%
3336 \csname if\@tempa\endcsname
3337 \MT@error{The `copyfonts' option doesn't work with luatex}
3338 {Use pdftex instead.}%
3339 \fi
3340 }
3341 }{
3342 (lua)
3343 \MT@def@bool@opt{copyfonts}{%
3344 \csname if\@tempa\endcsname
3345 \MT@gl@et\MT@copy@font\MT@copy@font@
3346 \else
3347 \MT@gl@et\MT@copy@font\relax
3348 \fi
3349 }
3350 (lua) }
3351 }{
3352 \MT@def@bool@opt{copyfonts}{%
3353 \csname if\@tempa\endcsname
3354 \MT@error{The pdftex version you are using is too old\MessageBreak
3355 to use the `copyfonts' option}{Upgrade pdftex.}%
3356 \fi
3357 }
3358 }

```

`final` is the opposite to `draft`.

```

3359 \MT@def@bool@opt{final}{%
3360 \csname if\@tempa\endcsname
3361 \MT@draftfalse
3362 \else
3363 \MT@drafttrue
3364 \fi
3365 }

```



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

```
3366 \define@key{MT}{verbose}[true]{%
3367   \let\MT@vinfo\MT@info@n1
3368   \def\@tempa{#1}%
3369   \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
3370   \MT@ifstreq\@tempa{errors}{%
3371     \let\MT@warning \MT@warn@err
3372     \let\MT@warning@n1\MT@warn@err
3373   }{%
3374     \let\MT@vinfo\@gobble
```

Cast warnings to the winds.

```
3375   \MT@ifstreq\@tempa{silent}{%
3376     \let\MT@warning \MT@info
3377     \let\MT@warning@n1\MT@info@n1
3378   }{%
3379     \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
3380   }%
3381 }%
3382 }%
3383 }
```

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

```
3384 </package>
3385 <plain>\ifnum\MT@plain<2
3386 \MT@map@clist@n{%
3387 <package> stretch,shrink,step,%
3388 letterspace}{%
3389 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
3390 \def\@tempa{##1 }%
```

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

```
3391 \MT@ifint\@tempa
3392 {\MT@edef@n{MT@#1}{\@tempa}}%
3393 {\MT@optwarn@nan{##1}{#1}}%
3394 }%
3395 }
3396 <plain>\fi
3397 <*package>
```

factor will define the protrusion factor only.

```
3398 \define@key{MT}{factor}[\MT@factor@default]{%
3399 \def\@tempa{#1 }%
3400 \MT@ifint\@tempa
3401 {\edef\MT@pr@factor{\@tempa}}
3402 {\MT@optwarn@nan{#1}{factor}}%
3403 }
```

Unit for protrusion codes.

```
3404 \define@key{MT}{unit}[character]{%
3405 \def\@tempa{#1}%
3406 \MT@ifstreq\@tempa{character}\relax{%
3407   \MT@ifdimen\@tempa
3408   {\let\MT@pr@unit\@tempa}%
3409   {\MT@warning@n1{~\@tempa' is not a dimension.\MessageBreak
3410     Ignoring it and setting values relative to\MessageBreak
3411     character widths}}%
3412 }%
3413 }
```

### 14.4.2 Reading the Configuration File

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as  $\text{\TeX}$  systems are switching to the pdf $\text{\TeX}$  engine even for DVI output, so that the user might not even be aware of the fact that she's running pdf $\text{\TeX}$ .)

```
3414 \MT@protrusiontrue
3415 \ifnum\pdfoutput<\@ne \else
```

Also, we only enable expansion by default if pdf $\text{\TeX}$  can expand the fonts automatically.

```
3416 \MT@requires@pdftex4{
3417   \MT@expansiontrue
3418   \MT@autottrue
3419 } \relax
3420 \fi
```

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

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

```
3421 \define@key{MT}{config}[] {\relax}
3422 \def\MT@get@config#1config=#2,#3\@nil{%
3423   \MT@ifempty{#2}%
3424   {\def\MT@config@file{\MT@MT.cfg}}%
3425   {\def\MT@config@file{#2.cfg}}%
3426 }
3427 \expandafter\expandafter\expandafter\MT@get@config
3428 \curname opt@ \@currname.\@currentx\endcurname,config=\@nil
```

Load the file.

```
3429 \IfFileExists{\MT@config@file}{%
3430   \MT@info@nl{Loading configuration file \MT@config@file}%
3431   \MT@begin@catcodes
3432   \let\MT@begin@catcodes\relax
3433   \let\MT@end@catcodes\relax
3434   \let\MT@curr@file\MT@config@file
3435   \input{\MT@config@file}%
3436   \endgroup
3437 }{\MT@warning@nl{%
3438   Could not find configuration file ` \MT@config@file'!\MessageBreak
3439   This will almost certainly cause undesired results.\MessageBreak
3440   Please fix your installation}%
3441 }
```

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

```
3442 \MT@map@clist@c\MT@features{%
3443   \MT@ifdefined@n@TF{MT@default@#1@set}\relax
3444   {\MT@gdef@n{MT@default@#1@set}{all}}%
3445 }
```

`\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).

```
3446 \def\MT@check@active@set#1{%
3447   \MT@ifdefined@n@TF{MT@#1@setname}{%
3448     \MT@info@nl{Using \@nameuse{MT@abbr@#1} set ` \@nameuse{MT@#1@setname}' }%
3449   }%
```

```

3450 \global\MT@let@nn{MT@#1@setname}{MT@default@#1@set}%
3451 \MT@info@n1{Using default \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}'}%
3452 }%
3453 }

```

### 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.

```

3454 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
3455   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3456   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
3457 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook

```

### 14.4.4 Changing Options Later

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

```

3458 \def\microtypesetup{\setkeys{MT}}
3459 \MT@addto@setup{\def\microtypesetup{\setkeys{MTX}}}
3460 \def\MT@define@optionX#1#2{%
3461   \define@key{MTX}{#1}[true]{%
3462     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3463     \MT@map@clist@n{##1}{%
3464       \KV@sp@def\MT@val{###1}%
3465       \MT@ifempty\MT@val\relax{%
3466         \@tempcnta=\m@ne
3467         \MT@ifstreq\MT@val{true}{%

```

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

```

3468 \MT@checksetup{#1}{%
3469 \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
3470 \MT@info{Enabling #1
3471 (level \number\csname MT@\@tempb @level\endcsname)}%
3472 }%
3473 }{%
3474 \MT@ifstreq\MT@val{false}{%
3475 \@tempcnta=\z@
3476 \MT@info{Disabling #1}%
3477 }{%
3478 \MT@ifstreq\MT@val{compatibility}{%
3479 \MT@checksetup{#1}{%
3480 \@tempcnta=\@ne
3481 \MT@let@nc{MT@\@tempb @level}\@ne
3482 \MT@info{Setting #1 to level 1}%
3483 }%
3484 }{%
3485 \MT@ifstreq\MT@val{nocompatibility}{%
3486 \MT@checksetup{#1}{%
3487 \@tempcnta=\tw@
3488 \MT@let@nc{MT@\@tempb @level}\tw@
3489 \MT@info{Setting #1 to level 2}%
3490 }%
3491 }{\MT@error{Value `MT@val' for key `#1' not recognised}
3492 {Use any of `true', `false', `compatibility' or
3493 `nocompatibility'.}%
3494 }%
3495 }%
3496 }%
3497 }%
3498 \ifnum\@tempcnta>\m@ne
3499 #2\@tempcnta\relax
3500 \fi
3501 }%
3502 }%
3503 }%
3504 }

```

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

```

3505 \def\MT@checksetup#1{%
3506 \csname ifMT@#1\endcsname
3507 \expandafter\@firstofone
3508 \else
3509 \MT@error{You cannot enable #1 if it was disabled\MessageBreak
3510 in the package options}{Load microtype with #1 enabled.}%
3511 \expandafter\@gobble
3512 \fi
3513 }

3514 \MT@define@optionX{protrusion}\pdfprotrudechars
3515 \MT@define@optionX{expansion}\pdfadjustspacing

```

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

```

3516 \MT@requires@pdftex6{
3517 \def\MT@define@optionX@#1#2{%
3518 \define@key{MTX}{#1}[true]{%
3519 \MT@map@clist@n{##1}{%
3520 \KV@sp@def\MT@val{###1}%

```

```

3521 \MT@ifempty\MT@val\relax{%
3522 \@tempcnta=\m@ne
3523 \MT@ifstreq\MT@val{true}{%
3524 \MT@checksetup{#1}{%
3525 \@tempcnta=\@ne
3526 \MT@info{Enabling #1}%
3527 }%
3528 }{%
3529 \MT@ifstreq\MT@val{false}{%
3530 \@tempcnta=\z@
3531 \MT@info{Disabling #1}%
3532 }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
3533 {Use either `true' or `false'}}%
3534 }%
3535 }%
3536 \ifnum\@tempcnta>\m@ne
3537 #2\relax
3538 \fi
3539 }%
3540 }%
3541 }%
3542 }

3543 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\relax\else
3544 \let\MT@tracking\MT@tracking@\fi}
3545 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
3546 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3547 \pdfappendkern \@tempcnta}
3548 }{

```

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

```

3549 \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
3550 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
3551 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
3552 }
3553 \define@key{MTX}{activate}[true]{%
3554 \setkeys{MTX}{protrusion={#1}}%
3555 \setkeys{MTX}{expansion={#1}}%
3556 }

```

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

```

3557 \let\MT@saved@setupfont\MT@setupfont
3558 \define@key{MTX}{disable}[]{}%
3559 \MT@info{Inactivate `~\MT@MT' package}%
3560 \let\MT@setupfont\relax
3561 }
3562 \define@key{MTX}{enable}[]{}%
3563 \MT@info{Reactivate `~\MT@MT' package}%
3564 \let\MT@setupfont\MT@saved@setupfont
3565 }
3566 /package

```

#### 14.4.5 Processing the Options

`\MT@ProcessOptionsWithKV`    Parse options.

```

3567 plain \ifnum\MT@plain<2
3568 \def\MT@ProcessOptionsWithKV#1{%
3569 \let\@tempc\relax
3570 \let\MT@temp\@empty
3571 plain \ifnum\MT@plain=0

```

```

3572 \MT@map@clist@{\classoptionslist{%
3573   \def\CurrentOption{##1}%
3574   \MT@ifdefined@n@T{KV@#1@}\expandafter\MT@getkey\CurrentOption=\@nil}{%
3575     \edef\MT@temp{\MT@temp,\CurrentOption,}%
3576     \expandtwoargs\@removeelement\CurrentOption
3577     \@unusedoptionlist\@unusedoptionlist
3578   }%
3579 }%
3580 \edef\MT@temp{\noexpand\setkeys{#1}%
3581   {\MT@temp\optionlist{\@currname.\@currentt}}}%

```

plain can handle package options.

```

3582 <plain>
3583 \else
3584   \edef\MT@temp{\noexpand\setkeys{#1}%
3585     {\csname usepkg@options@usepkg@pkg\endcsname}}%
3586 \fi
3587 </plain>
3588 \MT@temp
3589 \MT@clear@options
3590 }

```

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

```

3591 \def\MT@getkey#1=#2\@nil{#1}
3592 \MT@ProcessOptionsWithKV{MT}
3593 <plain>\fi
3594 <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).

```

3595 \MT@addto@setup{%
3596 \ifMT@draft

```

We disable most of what we've just defined in the 3596 lines above if we are running in draft mode.

```

3597 \MT@warning@nl{\`draft' option active.\MessageBreak
3598   Disabling all micro-typographic extensions.\MessageBreak
3599   This might lead to different line and page breaks}
3600 \let\MT@setupfont\relax
3601 \renewcommand*\LoadMicrotypeFile[1]{}
3602 \renewcommand*\microtypesetup[1]{}
3603 \renewcommand*\microtypecontext[1]{}
3604 \renewcommand*\lsstyle{}
3605 \else

```

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

```

3606 \ifnum\pdfoutput<\@ne
3607   \ifMT@opt@expansion \else
3608     \MT@expansionfalse
3609   \fi
3610 \fi

```

pdf<sub>T</sub><sub>E</sub>X can create DVI output, too. However, both the DVI viewer and dvips need to find actual fonts. Therefore, expansion will only work if the fonts for different degrees of expansion are readily available.

Some packages depend on the value of \pdfoutput and will get confused if it is changed after they have been loaded. These packages are, among others: color, graphics, hyperref, crop, contour, pstricks and, as a matter of course, ifpdf.

Instead of testing for each package (that's not our job), we only say that it was microtype that changed it. This must be sufficient!

```
3611 \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3612 \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
```

Fix the font sets.

```
3613 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
```

Protrusion.

```
3614 \ifMT@protrusion
3615 \edef\MT@active@features{\MT@active@features,pr}
3616 \pdfprotrudechars\MT@pr@level
3617 \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
3618 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3619 factor: \number\MT@pr@factor\fi
3620 \ifx\MT@pr@unit@empty \else,\MessageBreak unit: \MT@pr@unit\fi}
3621 \MT@check@active@set{pr}
3622 \else
3623 \let\MT@protrusion\relax
3624 \MT@info@nl{No character protrusion}
3625 \fi
```

Expansion.

```
3626 \ifMT@expansion
```

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

```
3627 \ifnum\MT@stretch=\m@ne
3628 \let\MT@stretch\MT@stretch@default
3629 \fi
```

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

```
3630 \ifnum\MT@shrink=\m@ne
3631 \let\MT@shrink\MT@stretch
3632 \fi
```

If step has not been specified, we will set it to  $\min(\text{stretch}, \text{shrink})/5$ , rounded off, minimum value 1.

```
3633 \ifnum\MT@step=\m@ne
3634 \ifnum\MT@stretch>\MT@shrink
3635 \ifnum\MT@shrink=\z@
3636 \@tempcnta=\MT@stretch
3637 \else
3638 \@tempcnta=\MT@shrink
3639 \fi
3640 \else
3641 \ifnum\MT@stretch=\z@
3642 \@tempcnta=\MT@shrink
3643 \else
3644 \@tempcnta=\MT@stretch
3645 \fi
3646 \fi
3647 \divide\@tempcnta 5\relax
3648 \else
3649 \@tempcnta=\MT@step
3650 \ifnum\@tempcnta=\z@
3651 \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
3652 Setting it to one}
3653 \fi
3654 \fi
3655 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
```

3656     \edef\MT@step{\number\@tempcnta\space}  
\MT@auto     Automatic expansion of the font? This new feature of pdf $\TeX$  1.20 makes the *l<sup>A</sup>T<sub>E</sub>X* programme really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdf $\TeX$ ).

3657     \let\MT@auto\empty  
3658     \ifMT@auto  
3659     \MT@requires@pdftex4{%

We turn off automatic expansion if output mode is DVI.

3660     \ifnum\pdfoutput<\@ne  
3661     \ifMT@opt@auto  
3662     \MT@error{%  
3663     Automatic font expansion only works for PDF output.\MessageBreak  
3664     However, you are creating a DVI file}  
3665     {If you have created expanded fonts instances, remove ‘auto’ from  
3666     \MessageBreak the package options. Otherwise, you have to switch  
3667     off expansion\MessageBreak completely.}  
3668     \fi  
3669     \MT@autofalse  
3670    \else  
3671     \def\MT@auto{autoexpand}  
3672    \fi

Also, if pdf $\TeX$  is too old.

3673     }{%  
3674     \MT@error{%  
3675     The pdf $\TeX$  version you are using is too old for\MessageBreak  
3676     automatic font expansion}%  
3677     {If you have created expanded fonts instances, remove ‘auto’ from\MessageBreak  
3678     the package options. Otherwise, you have to switch off expansion\MessageBreak  
3679     completely, or upgrade pdf $\TeX$  to version 1.20 or newer.}  
3680     \MT@autofalse  
3681     \def\MT@auto{1000 }  
3682    }  
3683    \else

No automatic expansion.

3684     \MT@requires@pdftex4\relax{  
3685     \def\MT@auto{1000 }  
3686    }  
3687    \fi

Choose the appropriate macro for selected expansion.

3688    \ifMT@selected  
3689     \let\MT@set@ex@codes\MT@set@ex@codes@s  
3690    \else  
3691     \let\MT@set@ex@codes\MT@set@ex@codes@n  
3692    \fi

Filter out stretch=0, shrink=0, since it would result in a pdf $\TeX$  error.

3693    \ifnum\MT@stretch=\z@  
3694    \ifnum\MT@shrink=\z@  
3695     \MT@warning@nl{%  
3696     Both the stretch and shrink limit are set to zero.\MessageBreak  
3697     Disabling font expansion}  
3698     \MT@expansionfalse  
3699    \fi  
3700    \fi  
3701    \fi  
3702    \ifMT@expansion  
3703     \edef\MT@active@features{\MT@active@features,ex}%



```

3704 \pdfadjustspacing\MT@ex@level
3705 \MT@info@n1{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
3706 (level \number\MT@ex@level),\MessageBreak
3707 stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3708 step: \number\MT@step, \ifMT@selected\else non-\fi selected%
3709 \ifx\MT@copy@font\relax\else .\MessageBreak
3710 Set up for expansion contexts (using font copies)\fi}

```

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

```

3711 \def\MT@check@step#1{%
3712 \tempcnta=\csname MT@#1\endcsname
3713 \divide\tempcnta \MT@step
3714 \multiply\tempcnta \MT@step
3715 \ifnum\tempcnta=\csname MT@#1\endcsname\else
3716 \MT@warning@n1{The #1 amount is not a multiple of step.\MessageBreak
3717 The effective maximum #1 is \the\tempcnta\space
3718 (step \number\MT@step)}
3719 \fi
3720 }
3721 \MT@check@step{stretch}
3722 \MT@check@step{shrink}
3723 \MT@check@active@set{ex}

```

Inside `\showhyphens`, font expansion should be disabled.

```

3724 \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3725 \color@begingroup\everypar{}\parfillskip\z@skip
3726 \hsize\maxdimen\normal font\pretolerance\m@ne\tolerance\m@ne
3727 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

```

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

```

3728 \gdef\showhyphens#1{\setbox0\vbox{%
3729 \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3730 \hsize\maxdimen\normal font\pretolerance\m@ne\tolerance\m@ne
3731 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
3732 \else
3733 \let\MT@expansion\relax
3734 \MT@info@n1{No font expansion}
3735 \fi
3736 }
3737 \MT@requires@pdftex6{
3738 \<package>
3739 \MT@addto@setup{%
3740 \<package>

```

Tracking, spacing and kerning.

```

3741 \ifMT@tracking
3742 \<lua>
3743 \MT@requires@luatex{
3744 \MT@error{Tracking doesn't currently work with luatex}{Use pdftex.}%
3745 \let\MT@tracking\relax
3746 }
3747 \</lua>
3748 \edef\MT@active@features{\MT@active@features,tr}
3749 \MT@info@n1{Tracking enabled}
3750 \MT@check@active@set{tr}

```

Enable protrusion for compensation at the line edges.

```

3751 \ifMT@protrusion\else\pdfprotrudechars\@ne\fi
3752 \<lua> }
3753 \else
3754 \let\MT@tracking\relax
3755 \MT@info@n1{No tracking}

```

```

3756 \fi
3757 \ifMT@spacing
3758   \edef\MT@active@features{\MT@active@features,sp}
3759   \pdfadjustinterwordglue\@ne
3760   \MT@info@nl{Adjustment of interword spacing enabled}
3761   \MT@check@active@set{sp}
3762 \else
3763   \let\MT@spacing\relax
3764   \MT@info@nl{No adjustment of interword spacing}
3765 \fi
3766 \ifMT@kerning
3767   \edef\MT@active@features{\MT@active@features,kn}
3768   \pdfprependkern\@ne
3769   \pdfappendkern\@ne
3770   \MT@info@nl{Adjustment of character kerning enabled}
3771   \MT@check@active@set{kn}
3772 \else
3773   \let\MT@kerning\relax
3774   \MT@info@nl{No adjustment of character kerning}
3775 \fi
3776 </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.

```

3777 \ifnum\pdfoutput<\@ne
3778   \def\MT@warn@tracking@DVI{%
3779     \MT@warning@nl{%
3780       You are using tracking/letterspacing in DVI mode.\MessageBreak
3781       This will probably not work, unless the post-\MessageBreak
3782       processing program (dvips, dvipdfm(x), ...) is\MessageBreak
3783       able to create the virtual fonts on the fly}%
3784     \MT@glet\MT@warn@tracking@DVI\relax
3785   }
3786 \else
3787   \def\MT@warn@tracking@DVI{%
3788     \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
3789     \MT@glet\MT@warn@tracking@DVI\relax
3790   }
3791 \fi
3792 \ifnum\MT@letterspace=\m@ne
3793   \let\MT@letterspace\MT@letterspace@default
3794 \else
3795   \MT@ls@too@large\MT@letterspace
3796 \fi
3797 }

```

If pdfTeX is too old, we disable tracking, spacing and kerning, and throw an error message.

```

3798 <*package>
3799 {}
3800 \MT@addto@setup{%
3801   \ifMT@tracking
3802     \MT@error{Tracking only works with pdftex version 1.40\MessageBreak
3803       or newer. Switching it off}{Upgrade pdftex.}%
3804   \else
3805     \MT@info@nl{No tracking (pdftex too old)}
3806   \fi
3807   \ifMT@spacing
3808     \MT@error{Adjustment of interword spacing only works with\MessageBreak
3809       pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%

```

```

3810 \else
3811 \MT@info@n{No adjustment of interword spacing (pdftex too old)}
3812 \fi
3813 \ifMT@kerning
3814 \MT@error{Character kerning only works with\MessageBreak
3815 pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
3816 \else
3817 \MT@info@n{No adjustment of character kerning (pdftex too old)}
3818 \fi
3819 }
3820 }

```

Warning if `\nonfrenchspacing` is active, since space factors will be ignored with `\pdfadjustinterwordglue>0`. Why 1500? Because some packages redefine `\frenchspacing`. See the c.t.t thread ‘`\frenchspacing` with AMS packages and babel’, started by Philipp Lehman on 16 August 2005: [ddtbaj\\$rob\\$1@online.de](mailto:ddtbaj$rob$1@online.de).

```

3821 \MT@requires@pdftex6{
3822 \AtBeginDocument{%
3823 \ifMT@spacing
3824 \ifMT@babel \else
3825 \ifnum\scode`. > 1500
3826 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
3827 \MT@warning@n{%
3828 \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3829 interword spacing will disable it. You might want\MessageBreak
3830 to add \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
3831 to your preamble}%
3832 }%
3833 \fi
3834 \fi
3835 \fi
3836 }
3837 }\relax

```

`\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```

3838 \MT@requires@pdftex5{
3839 \MT@addto@setup{%
3840 \ifMT@noligatures \else
3841 \let\MT@noligatures\relax
3842 \fi
3843 }
3844 }\relax

```

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

```

3845 \MT@addto@setup{%
3846 \ifx\MT@active@features\empty \else
3847 \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3848 \fi
3849 \MT@documenttrue
3850 }

```

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

```

3851 \def\MT@set@babel@context#1{%
3852 \MT@ifdefined@n@TF{MT@babel@#1}{%
3853 \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
3854 \expandafter\MT@exp@one@n\expandafter\microtypecontext
3855 \csname MT@babel@#1\endcsname
3856 }{%
3857 \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%

```

```
3858 }%
3859 }
```

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

```
3860 \ifpackageloaded{babel}{
3861   \def\MT@shorthandoff#1#2{%
3862     \MT@info@nl{Switching off #1 babel's active characters (#2)}%
3863     \shorthandoff{#2}}
3864 }{
3865   \def\MT@shorthandoff#1#2{%
3866     \MT@error{You must load `babel' before `MT@MT'}
3867     {Otherwise, `MT@MT' cannot switch off #1 babel's\MessageBreak
3868       active characters.}}
3869 }
```

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

```
3870 \MT@addto@setup{%
3871   \ifMT@babel
3872     \ifpackageloaded{babel}{%
3873       \MT@info@nl{Redefining babel's language switching commands}
3874       \let\MT@orig@select@language\select@language
3875       \def\select@language#1{%
3876         \MT@orig@select@language{#1}%
3877         \MT@set@babel@context{#1}%
3878       }
3879       \let\MT@orig@foreign@language\foreign@language
3880       \def\foreign@language#1{%
3881         \MT@orig@foreign@language{#1}%
3882         \MT@set@babel@context{#1}%
3883       }
3884     \ifMT@kerning
```

Disable French babel's active characters.

```
3885     \MT@if@false
3886     \MT@with@babel@and@T{french} \MT@if@true
3887     \MT@with@babel@and@T{frenchb} \MT@if@true
3888     \MT@with@babel@and@T{français}\MT@if@true
3889     \MT@with@babel@and@T{canadien}\MT@if@true
3890     \MT@with@babel@and@T{acadian} \MT@if@true
3891     \ifMT@if@MT@shorthandoff{French}{:;!}\fi
```

Disable Turkish babel's active characters.

```
3892     \MT@if@false
3893     \MT@with@babel@and@T{turkish} \MT@if@true
3894     \ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi
3895     \fi
```

In case babel was loaded before microtype:

```
3896     \MT@set@babel@context\language
3897   }{%
3898     \MT@warning@nl{You did not load the babel package.\MessageBreak
3899     The `babel' option won't have any effect}
3900   }
3901 \fi
3902 }
```

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

```
3903 \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.

```
3904 \AtBeginDocument\selectfont
```

```

\MT@curr@file    This is the current file (hopefully with the correct extension).
3905 \edef\MT@curr@file{\jobname.tex}

    Restore catcodes.
3906 </package>
3907 \MT@restore@catcodes

    That was that.
3908 </package|letterspace>

```

## 15 Configuration Files

Let's now write the font configuration files.

```

3909 <*config>
3910

```

### 15.1 Font Sets

We first declare some sets in the main configuration file.

```

3911 <*m-t>
3912 %%% -----
3913 %%% FONT SETS
3914
3915 \DeclareMicrotypeSet{all}
3916 { }
3917
3918 \DeclareMicrotypeSet{allmath}
3919 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
3920
3921 \DeclareMicrotypeSet{alltext}
3922 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
3923
3924 \DeclareMicrotypeSet{basicmath}
3925 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
3926   family   = {rm*,sf*},
3927   series    = {md*},
3928   size      = {normalsize,footnotesize,small,large}
3929 }
3930
3931 \DeclareMicrotypeSet{basictext}
3932 { encoding = {OT1,T1,LY1,OT4,QX,T5},
3933   family   = {rm*,sf*},
3934   series    = {md*},
3935   size      = {normalsize,footnotesize,small,large}
3936 }
3937
3938 \DeclareMicrotypeSet{smallcaps}
3939 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3940   shape     = {sc}
3941 }
3942
3943 \DeclareMicrotypeSet{footnotesize}
3944 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3945   size      = {-small}
3946 }
3947
3948 \DeclareMicrotypeSet{scriptsize}
3949 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},

```

```

3950     size      = {-footnotesize}
3951   }
3952
3953 \DeclareMicrotypeSet{normalfont}
3954   { font = */*/*/*/* }
3955

```

The default sets.

```

3956 %%% -----
3957 %%% DEFAULT SETS
3958
3959 \DeclareMicrotypeSetDefault[protrusion]{alltext}
3960 \DeclareMicrotypeSetDefault[expansion]{basictext}
3961 \DeclareMicrotypeSetDefault[spacing]{basictext}
3962 \DeclareMicrotypeSetDefault[kerning]{alltext}
3963 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
3964

```

## 15.2 Font Variants and Aliases

```

3965 %%% -----
3966 %%% FONT VARIANTS AND ALIASES
3967

```

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

```

3968 \DeclareMicrotypeVariants{x,j,w,a,d,0,1}
3969

```

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.

```

3970 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
3971 \DeclareMicrotypeAlias{aer}{cmr} % ae
3972 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
3973 \DeclareMicrotypeAlias{cmor}{cmr} % eco
3974 \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).

```

3975 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
3976 \DeclareMicrotypeAlias{qpl}{ppl} % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

```

The 'FPL Neu' fonts, a 're-implementation' of Palatino.

```

3977 \DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
3978 \DeclareMicrotypeAlias{fp9j}{pplj} % "
3979 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
3980 \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.

```

3981 \DeclareMicrotypeAlias{eur}{eur} % Euler VM
3982 \DeclareMicrotypeAlias{zeus}{eus} % "

```

MicroPress's Charter version (chmath).

```

3983 \DeclareMicrotypeAlias{chr}{bch} % CH Math

```

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```

3984 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
3985 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond

```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```

3986 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch

```

Euro symbol fonts, to save some files.

```

3987 \DeclareMicrotypeAlias{zpeu}{zpeu} % Adobe Euro sans -> serif
3988 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
3989 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
3990

```

## 15.3 Interaction with babel

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

```

3991 %%% -----
3992 %%% INTERACTION WITH THE `babel' PACKAGE
3993
3994 \DeclareMicrotypeBabelHook
3995   {english,UKenglish,british,USenglish,american}
3996   {kerning=, spacing=nonfrench}
3997
3998 \DeclareMicrotypeBabelHook
3999   {french,français,acadian,canadien}
4000   {kerning=french, spacing=}
4001
4002 \DeclareMicrotypeBabelHook
4003   {turkish}
4004   {kerning=turkish, spacing=}
4005

```

## 15.4 Note on Admissible Characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```

\ : \textbackslash
{ : \textbraceleft
} : \textbraceright
^ : \textasciicircum
% : \%
# : \#

```

Comma and equal sign must be guarded with braces ('{,}', '{=}') to keep keyval happy.

Character commands are allowed as far as they have been defined in the proper  $\text{\LaTeX}$  way, that is, when they have been assigned a slot in the font encoding with `\DeclareTextSymbol` or `\DeclareTextComposite`. Characters defined via `\chardef` are also possible.

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.

```

4022 \DeclareCharacterInheritance
4023 { encoding = T1 }
4024 { A = {\^A,\^A,\^A,\^A,\^A,\r A,\k A,\u A},
4025   a = {\`a,\`a,\`a,\`a,\`a,\r a,\k a,\u a},
4026   C = {\^C,\c C,\v C},
4027   c = {\`c,\c c,\v c},
4028   D = {\v D,\DH},
4029   d = {\v d,\dj},
4030   E = {\^E,\^E,\^E,\^E,\^E,\k E,\v E},
4031   e = {\`e,\`e,\`e,\`e,\`e,\k e,\v e},
4032   f = {027}, % ff
4033   G = {\u G},
4034   g = {\u g},
4035   I = {\^I,\^I,\^I,\^I,\^I,\^I,\^I},
4036   i = {\`i,\`i,\`i,\`i,\`i,\`i},
4037   j = {\j},
4038   L = {\L,\^L,\v L},
4039   l = {\l,\l,\v l},
4040   N = {\^N,\^N,\v N},

```



```

4041 n = {\`n,\~n,\v n},
4042 O = {\0,\`0,\'0,\^0,\-0,\"0,\H 0},
4043 o = {\o,\`o,\'o,\^o,\-o,\"o,\H o},
4044 R = {\`R,\v R},
4045 r = {\`r,\v r},
4046 S = {\`S,\c S,\v S,\SS},
4047 s = {\`s,\c s,\v s},
4048 T = {\c T,\v T},
4049 t = {\c t,\v t},
4050 U = {\`U,\'U,\^U,\"U,\H U,\r U},
4051 u = {\`u,\'u,\^u,\"u,\H u,\r u},
4052 Y = {\`Y,\"Y},
4053 y = {\`y,\"y},
4054 Z = {\`Z,\.Z,\v Z},
4055 z = {\`z,\.z,\v z}

```

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

```

4056 % - = {127},
4057 }
4058

```

### 15.5.3 LY1

More characters: 008 (‘f’), 012 (‘fi’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```

4059 \DeclareCharacterInheritance
4060 { encoding = LY1 }
4061 { A = {\`A,\'A,\^A,\-A,\"A,\r A},
4062   a = {\`a,\'a,\^a,\-a,\"a,\r a},
4063   C = {\c C},
4064   c = {\c c},
4065   D = {\DH},
4066   E = {\`E,\'E,\^E,\"E},
4067   e = {\`e,\'e,\^e,\"e},
4068   f = {011}, % ff
4069   I = {\`I,\'I,\^I,\"I},
4070   i = {\`i,\'i,\^i,\"i,\i},
4071   L = {\L},
4072   l = {\l},
4073   N = {\-N},
4074   n = {\-n},
4075   O = {\`O,\'O,\^O,\-O,\"O,\O},
4076   o = {\`o,\'o,\^o,\-o,\"o,\o},
4077   S = {\v S},
4078   s = {\v s},
4079   U = {\`U,\'U,\^U,\"U},
4080   u = {\`u,\'u,\^u,\"u},
4081   Y = {\`Y,\"Y},
4082   y = {\`y,\"y},
4083   Z = {\v Z},
4084   z = {\v z}
4085 }
4086

```

### 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.

```

4087 \DeclareCharacterInheritance

```

```

4088 { encoding = OT4 }
4089 { A = {\k A},
4090   a = {\k a},
4091   C = {\'C},
4092   c = {\'c},
4093   E = {\k E},
4094   e = {\k e},
4095   f = {011}, % ff
4096   i = {\i},
4097   j = {\j},
4098   L = {\L},
4099   l = {\l},
4100   N = {\'N},
4101   n = {\'n},
4102   O = {\0,\'0},
4103   o = {\o,\'o},
4104   S = {\'S},
4105   s = {\'s},
4106   Z = {\'Z,\.Z},
4107   z = {\'z,\.z}
4108 }
4109

```

### 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.

```

4110 \DeclareCharacterInheritance
4111 { encoding = QX }
4112 { A = {\`A,\'A,\^A,\~A,\"A,\k A,\AA},
4113   a = {\`a,\'a,\^a,\~a,\"a,\k a,\aa},
4114   C = {\'C,\c C},
4115   c = {\'c,\c c},
4116   D = {\DH},
4117   E = {\`E,\'E,\^E,\"E,\k E},
4118   e = {\`e,\'e,\^e,\"e,\k e},
4119   f = {011}, % ff
4120   I = {\`I,\'I,\^I,\"I,\k I},
4121   i = {\`i,\'i,\^i,\"i,\k i,\i},
4122   j = {\j},
4123   L = {\L},
4124   l = {\l},
4125   N = {\'N,\~N},
4126   n = {\'n,\~n},
4127   O = {\0,\`0,\'0,\^0,\~0,\"0},
4128   o = {\o,\`o,\'o,\^o,\~o,\"o},
4129   S = {\'S,\c S,\v S},
4130   s = {\'s,\c s,\v s},
4131   T = {\c T},
4132   t = {\c t},
4133   U = {\`U,\'U,\^U,\"U,\k U},
4134   u = {\`u,\'u,\^u,\"u,\k u},
4135   Y = {\'Y,\"Y},
4136   y = {\'y,\"y},
4137   Z = {\'Z,\.Z,\v Z},
4138   z = {\'z,\.z,\v z},
4139   . = \textellipsis
4140 }
4141

```

---

<sup>13</sup> Thanks to Maciej Eder.

### 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.

```

4142 \DeclareCharacterInheritance
4143 { encoding = T5 }
4144 { A = {\`A,\`A,\~A,\h A,\d A,\^A,\u A,
4145       \`Acircumflex,\`Acircumflex,\~Acircumflex,\hAcircumflex,\dAcircumflex,
4146       \`Abreve,\`Abreve,\~Abreve,\hAbreve,\dAbreve},
4147   a = {\`a,\`a,\~a,\h a,\d a,\^a,\u a,
4148       \`acircumflex,\`acircumflex,\~acircumflex,\hacircumflex,\dacircumflex,
4149       \`abreve,\`abreve,\~abreve,\h\abreve,\d\abreve},
4150   D = {\DJ},
4151   d = {\dj},
4152   E = {\`E,\`E,\~E,\h E,\d E,\^E,
4153       \`Ecircumflex,\`Ecircumflex,\~Ecircumflex,\hEcircumflex,\dEcircumflex},
4154   e = {\`e,\`e,\~e,\h e,\d e,\^e,
4155       \`ecircumflex,\`ecircumflex,\~ecircumflex,\h\ecircumflex,\d\ecircumflex},
4156   I = {\`I,\`I,\~I,\h I,\d I},
4157   i = {\`i,\`i,\~i,\h i,\d i,\i},
4158   O = {\`O,\`O,\~O,\h O,\d O,\^O,\horn O,
4159       \`Ocircumflex,\`Ocircumflex,\~Ocircumflex,\hOcircumflex,\dOcircumflex,
4160       \`Ohorn,\`Ohorn,\~Ohorn,\hOhorn,\dOhorn},
4161   o = {\`o,\`o,\~o,\h o,\d o,\^o,\horn o,
4162       \`ocircumflex,\`ocircumflex,\~ocircumflex,\h\ocircumflex,\d\ocircumflex,
4163       \`ohorn,\`ohorn,\~ohorn,\hohorn,\d\ohorn},
4164   U = {\`U,\`U,\~U,\h U,\d U,\horn U,
4165       \`Uhorn,\`Uhorn,\~Uhorn,\hUhorn,\dUhorn},
4166   u = {\`u,\`u,\~u,\h u,\d u,\horn u,
4167       \`uhorn,\`uhorn,\~uhorn,\huhorn,\d\uhorn},
4168   Y = {\`Y,\`Y,\~Y,\h Y,\d Y},
4169   y = {\`y,\`y,\~y,\h y,\d y}
4170 }
4171
4172 </m-t>

```

### 15.5.7 Euro symbols

Make Euro symbols settings simpler.

```

4173 < *zpeu >
4174 \DeclareCharacterInheritance
4175 { encoding = U,
4176   family = {zpeu,zpeus,eurosans} }
4177 { E = 128 }
4178
4179 < /zpeu >
4180 < *mvs >
4181 \DeclareCharacterInheritance
4182 { encoding = OT1,
4183   family = mvs }
4184 { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
4185

```

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.

```

4186 \DeclareCharacterInheritance
4187 { encoding = U,
4188   family = mvs }
4189 { 164 = {099,100,101} }

```

```
4190
4191 </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.

```
4192 (*m-t)
4193 %%% -----
4194 %%% TRACKING/LETTERSPACING
4195
4196 \SetTracking
4197 [ name          = default,
4198   no ligatures = {f} ]
4199 { encoding      = {OT1,T1,LY1,OT4,QX} }
4200 { }
4201
```

## 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).

```
4202 %%% -----
4203 %%% EXPANSION
4204
4205 \SetExpansion
4206 [ name          = default      ]
4207 { encoding      = {OT1,OT4,QX,T1,LY1} }
4208 {
4209   A = 500,      a = 700,
4210   \AE = 500,    \ae = 700,
4211   B = 700,      b = 700,
4212   C = 700,      c = 700,
4213   D = 500,      d = 700,
4214   E = 700,      e = 700,
4215   F = 700,
4216   G = 500,      g = 700,
4217   H = 700,      h = 700,
4218   K = 700,      k = 700,
4219   M = 700,      m = 700,
4220   N = 700,      n = 700,
4221   O = 500,      o = 700,
4222   \OE = 500,    \oe = 700,
4223   P = 700,      p = 700,
4224   Q = 500,      q = 700,
4225   R = 700,
4226   S = 700,      s = 700,
4227   U = 700,      u = 700,
4228   W = 700,      w = 700,
4229   Z = 700,      z = 700,
4230   2 = 700,
4231   3 = 700,
4232   6 = 700,
4233   8 = 700,
4234   9 = 700
4235 }
4236
```

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

```

4237 \SetExpansion
4238 [ name      = T5 ]
4239 { encoding = T5 }
4240 {
4241   A = 500,      a = 700,
4242   B = 700,      b = 700,
4243   C = 700,      c = 700,
4244   D = 500,      d = 700,
4245   E = 700,      e = 700,
4246   F = 700,
4247   G = 500,      g = 700,
4248   H = 700,      h = 700,
4249   K = 700,      k = 700,
4250   M = 700,      m = 700,
4251   N = 700,      n = 700,
4252   O = 500,      o = 700,
4253   P = 700,      p = 700,
4254   Q = 500,      q = 700,
4255   R = 700,
4256   S = 700,      s = 700,
4257   U = 700,      u = 700,
4258   W = 700,      w = 700,
4259   Z = 700,      z = 700,
4260   2 = 700,
4261   3 = 700,
4262   6 = 700,
4263   8 = 700,
4264   9 = 700
4265 }
4266
4267 (/m-t)

```

## 15.8 Character Protrusion

```

4268 %%% -----
4269 %%% PROTRUSION
4270

```

For future historians, Hàn Thế Thành's original settings (from `protcode.tex`, converted to microtype notation).

```

\SetProtrusion
[ name      = thanh ]
{ encoding = OT1 }
{
  A = {50,50},
  F = { ,50},
  J = {50, },
  K = { ,50},
  L = { ,50},
  T = {50,50},
  V = {50,50},
  W = {50,50},
  X = {50,50},
  Y = {50,50},
  k = { ,50},
  r = { ,50},
  t = { ,50},
  v = {50,50},
  w = {50,50},
  x = {50,50},
  y = {50,50},

```

```

. = { ,700}, {,}= { ,700},
: = { ,500}, ; = { ,500},
! = { ,200}, ? = { ,200},
( = {50, }, ) = { ,50},
- = { ,700},
\textendash = { ,300}, \textemdash = { ,200},
\textquoteleft = {700, }, \textquoteright = { ,700},
\textquotedblleft = {500, }, \textquotedblright = { ,500}
}

```

### 15.8.1 Default

The default settings always use the most moderate value.

```

4271 <*cfg-t>
4272 \SetProtrusion
4273 <m-t> [ name = default ]

```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```

4274 <bch> [ name = bch-default ]

```

- Bitstream Letter Gothic (blg)

```

4275 <blg> [ name = blg-default ]

```

- Computer Modern Roman (cmr)

```

4276 <cmr> [ name = cmr-default ]

```

- Adobe Garamond (pad, padx, padj)

```

4277 <pad> [ name = pad-default ]

```

- Minion<sup>14</sup> (pmnx, pmnj)

```

4278 <pmn> [ name = pmnj-default ]

```

- Palatino (ppl, pplx, pplj)

```

4279 <ppl> [ name = ppl-default ]

```

- Times (ptm, ptmx, ptmj)

```

4280 <ptm> [ name = ptm-default ]

```

- URW Garamond (ugm)

```

4281 <ugm> [ name = ugm-default ]
4282 <m-t> { encoding = OT1 }
4283 <cmr> { }
4284 <bch|blg|pad|pmn|ugm> { encoding = OT1,
4285 <ppl|ptm> { encoding = {OT1,OT4},
4286 <bch> family = bch }
4287 <blg> family = blg }
4288 <pad> family = {pad,padx,padj} }
4289 <pmn> family = pmnj }
4290 <ppl> family = {ppl,pplx,pplj} }
4291 <ptm> family = {ptm,ptmx,ptmj} }
4292 <ugm> family = ugm }
4293 {

```

14 Contributed by Harald Harders ([h.harders@tu-bs.de](mailto:h.harders@tu-bs.de)).

```

4294 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm>    A = {50,50},
4295 <ugm>    A = {50,100},
4296 <m-t|pad|ptm>    \AE = {50, },
4297 <ugm>    \AE = {150,50},
4298 <ugm>    B = { ,50},
4299 <bch|pad|pmn|ugm>    C = {50, },
4300 <bch|pad|pmn>    D = { ,50},
4301 <ugm>    D = { ,70},
4302 <ugm>    E = { ,50},
4303 <m-t|bch|cmr|pad|pmn|ptm>    F = { ,50},
4304 <ugm>    F = { ,70},
4305 <bch|pad|pmn>    G = {50, },
4306 <ugm>    G = {50,50},
4307 <blg>    I = {150,150},
4308 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    J = {50, },
4309 <bch|blg>    J = {100, },
4310 <!blg>    K = { ,50},
4311 <blg>    K = {50, },
4312 <m-t|bch|cmr|pad|pmn|ppl>    L = { ,50},
4313 <blg>    L = { ,150},
4314 <ptm>    L = { ,80},
4315 <ugm>    L = { ,120},
4316 <bch|pad|pmn|ugm>    O = {50,50},
4317 <pad|pmn>    \OE = {50, },
4318 <ugm>    \OE = {50,50},
4319 <blg>    P = { ,100},
4320 <ugm>    P = { ,50},
4321 <bch|pad|pmn>    Q = {50,70},
4322 <ugm>    Q = {50,50},
4323 <bch>    R = { ,50},
4324 <ugm>    R = { ,70},
4325 <m-t|bch|cmr|pad|pmn|ppl|ptm>    T = {50,50},
4326 <blg>    T = {100,100},
4327 <ugm>    T = {70,70},
4328 <m-t|bch|cmr|pad|pmn|ppl|ptm>    V = {50,50},
4329 <blg|ugm>    V = {70,70},
4330 <m-t|bch|cmr|pad|pmn|ppl|ptm>    W = {50,50},
4331 <ugm>    W = {70,70},
4332 <m-t|bch|cmr|pad|pmn|ppl|ptm>    X = {50,50},
4333 <ugm>    X = {50,70},
4334 <m-t|bch|cmr|pad|pmn|ppl>    Y = {50,50},
4335 <blg|ptm|ugm>    Y = {80,80},
4336 <ugm>    Z = {50,50},
4337 <blg>    f = {150,100},
4338 <blg>    i = {150,150},
4339 <blg>    j = {100,100},
4340 <m-t|bch|cmr|pad|pmn|ppl|ptm>    k = { ,50},
4341 <ugm>    k = { ,70},
4342 <blg>    l = {150,150},
4343 <pmn>    l = { ,50},
4344 <pad|ppl>    p = {50,50},
4345 <ugm>    p = { ,50},
4346 <pad|ppl>    q = {50, },
4347 <!blg>    r = { ,50},
4348 <blg>    r = {100, 80},
4349 <cmr|pad|pmn>    t = { ,70},
4350 <bch>    t = { ,50},
4351 <blg>    t = {150, 80},
4352 <ugm>    t = { ,100},
4353 <m-t|bch|cmr|pad|pmn|ppl|ptm>    v = {50,50},
4354 <blg>    v = {100,100},
4355 <ugm>    v = {50,70},
4356 <m-t|bch|cmr|pad|pmn|ppl|ptm>    w = {50,50},

```

```

4357 <ugm>      w = {50,70},
4358 <!blg>     x = {50,50},
4359 <blg>      x = {100,100},
4360 <m-t|bch|pad|pmn>    y = { ,50},
4361 <blg>      y = { 50,100},
4362 <cmr|ppl|ptm>    y = {50,70},
4363 <ugm>      y = { ,70},

4364 <cmr>      0 = { ,50},
4365 <m-t>      1 = {50,50},
4366 <bch|blg|pad|ptm|ugm>    1 = {150,150},
4367 <cmr>      1 = {100,200},
4368 <pmn>      1 = { ,50},
4369 <ppl>      1 = {100,100},
4370 <bch|cmr|pad|ugm>    2 = {50,50},
4371 <blg>      2 = { ,100},
4372 <bch|pmn>    3 = {50, },
4373 <cmr|pad|ugm>    3 = {50,50},
4374 <blg>      3 = {100, },
4375 <m-t|pad>    4 = {50,50},
4376 <bch>      4 = {100,50},
4377 <blg>      4 = {100, },
4378 <cmr|ugm>    4 = {70,70},
4379 <pmn>      4 = {50, },
4380 <ptm>      4 = {70, },
4381 <cmr>      5 = { ,50},
4382 <pad>      5 = {50,50},
4383 <bch>      6 = {50, },
4384 <cmr>      6 = { ,50},
4385 <pad>      6 = {50,50},
4386 <m-t>      7 = {50,50},
4387 <bch|pad|pmn|ugm>    7 = {50,80},
4388 <blg>      7 = {100,100},
4389 <cmr|ptm>    7 = {50,100},
4390 <ppl>      7 = { ,50},
4391 <cmr>      8 = { ,50},
4392 <bch|pad>    9 = {50,50},
4393 <cmr>      9 = { ,50},
4394 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    . = { ,700},
4395 <bch>      . = { ,600},
4396 <blg>      . = {400,500},
4397 <!blg>     {,}= { ,500},
4398 <blg>     {,}= {300,400},
4399 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    : = { ,500},
4400 <bch>      : = { ,400},
4401 <blg>      : = {300,400},
4402 <m-t|bch|pad|pmn|ptm>    ; = { ,300},
4403 <blg>      ; = {200,300},
4404 <cmr|ppl>    ; = { ,500},
4405 <ugm>      ; = { ,400},
4406 <!blg>     ! = { ,100},
4407 <blg>      ! = {200,200},
4408 <m-t|pad|pmn|ptm>    ? = { ,100},
4409 <bch|cmr|ppl|ugm>    ? = { ,200},
4410 <blg>      ? = {150,150},
4411 <pmn>      " = {300,300},
4412 <m-t|bch|cmr|pad|pmn|ppl>    @ = {50,50},
4413 <ptm>      @ = {100,100},
4414 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm>    ~ = {200,250},
4415 <ugm>      ~ = {300,350},
4416 <pad|ppl|ptm>    & = {50,100},
4417 <ugm>      & = { ,100},
4418 <m-t|cmr|pad|pmn>    \% = {50,50},

```



```

4419 <bch> \% = { ,50},
4420 <ppl|ptm> \% = {100,100},
4421 <ugm> \% = {50,100},
4422 <blg> \# = {100,100},
4423 <m-t|ppl|ptm|ugm> * = {200,200},
4424 <bch|pmn> * = {200,300},
4425 <blg> * = {150,200},
4426 <cmr|pad> * = {300,300},
4427 <m-t|cmr|ppl|ptm> + = {250,250},
4428 <bch> + = {150,250},
4429 <pad> + = {300,300},
4430 <blg|pmn> + = {150,200},
4431 <ugm> + = {250,300},
4432 <blg|ugm> {=} = {200,200},
4433 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
4434 <bch|ugm> ( = {200, }, ) = { ,200},
4435 <cmr|blg> ( = {300, }, ) = { ,300},
4436 <ppl> ( = {100, }, ) = { ,300},
4437 <bch|pmn> [ = {100, }, ] = { ,100},
4438 <blg> [ = {300,100}, ] = { ,300},

4439 <m-t|pad|pmn|ptm> / = {100,200},
4440 <bch> / = { ,200},
4441 <blg> / = {300,300},
4442 <cmr|ppl> / = {200,300},
4443 <ugm> / = {100,300},
4444 <m-t|ptm> - = {500,500},
4445 <bch|cmr|ppl> - = {400,500},
4446 <blg> - = {300,400},
4447 <pad> - = {300,500},
4448 <pmn> - = {200,400},
4449 <ugm> - = {500,600},
4450 <blg> <= {200,100}, >= {100,200},
4451 <blg> _ = {150,250},
4452 <blg> | = {250,250},
4453 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
4454 <bch> \textendash = {200,300}, \textendash = {150,250},
4455 <cmr> \textendash = {400,300}, \textendash = {300,200},
4456 <pad|ppl|ptm> \textendash = {300,300}, \textendash = {200,200},
4457 <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).

```

4458 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
4459 <blg> \textquoteleft = {400,600}, \textquoteright = {400,600},
4460 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
4461 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
4462 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
4463 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
4464 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300}
4465 <blg> \textquotedblright = {300,400}
4466 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600}
4467 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4468 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400}
4469 }
4470

```

Greek uppercase letters are in OT1 encoding only.

```

4471 (*cmr)
4472 \SetProtrusion
4473 [ name = cmr-OT1,
4474   load = cmr-default ]

```

```

4475 { encoding = {OT1,OT4},
4476   family   = cmr }
4477 {
4478   \AE = { 50, },
4479   "00 = { ,150}, % \Gamma
4480   "01 = {100,100}, % \Delta
4481   "02 = { 50, 50}, % \Theta
4482   "03 = {100,100}, % \Lambda
4483   "06 = { 50, 50}, % \Sigma
4484   "07 = {100,100}, % \Upsilon
4485   "08 = { 50, 50}, % \Phi
4486   "09 = { 50, 50} % \Psi

```

Remaining slots can be found in the source file.

```

4487 }
4488
4489 (/cmr)

```

T1 and LY1 encodings contain some more characters. The default list will be loaded first.

```

4490 \SetProtrusion
4491 <m-t> [ name      = T1-default,
4492 <bch>  [ name      = bch-T1,
4493 <blg>  [ name      = blg-T1,
4494 <cmr>  [ name      = cmr-T1,
4495 <pad>  [ name      = pad-T1,
4496 <pmn>  [ name      = pmn-T1,
4497 <ppl>  [ name      = ppl-T1,
4498 <ptm>  [ name      = ptm-T1,
4499 <ugm>  [ name      = ugm-T1,
4500 <m-t>  load      = default      ]
4501 <bch>  load      = bch-default ]
4502 <blg>  load      = blg-default ]
4503 <cmr>  load      = cmr-default ]
4504 <pad>  load      = pad-default ]
4505 <pmn>  load      = pmn-default ]
4506 <ppl>  load      = ppl-default ]
4507 <ptm>  load      = ptm-default ]
4508 <ugm>  load      = ugm-default ]
4509 <m-t> { encoding = {T1,LY1} }
4510 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4511 <blg|ptm|ugm> { encoding = {T1},
4512 <bch>    family   = bch }
4513 <blg>    family   = blg }
4514 <cmr>    family   = cmr }
4515 <pad>    family   = {pad,padx,padj} }
4516 <pmn>    family   = pmnj }
4517 <ppl>    family   = {ppl,pplx,pplj} }
4518 <ptm>    family   = {ptm,ptmx,ptmj} }
4519 <ugm>    family   = ugm }
4520 {
4521 <cmr>    \AE = {50, },
4522 <bch>    \OE = {50, },
4523 <pmn>    \TH = { ,50},
4524 <blg>    \v L = { ,250},
4525 <blg>    \v d = { ,250},
4526 <blg>    \v l = { ,250},
4527 <blg>    \v t = { ,250},
4528 <blg>    127 = {300,400},
4529 <blg>    156 = {100, }, % IJ
4530 <blg>    188 = { 80, 80}, % ij
4531 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},

```

```

4532 <cmr>      _ = {200,200},
4533 <ugm>      _ = {100,200},
4534 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
4535 <bch>      \textbackslash = {150,200},
4536 <blg>      \textbackslash = {250,300},
4537 <cmr|ppl> \textbackslash = {200,300},
4538 <ugm>      \textbackslash = {100,300},
4539 <ugm>      \textbar = {200,200},
4540 <blg>      \textendash = {300,300}, \textemdash = {150,150},
4541 <blg>      \textquotedbl = {300,400}, \textquotedblleft = {300,400},
4542 <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.

```

4543 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4544 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
4545 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4546 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4547 <blg> \guilsinglleft = {300,500}, \guilsinglright = {300,500},
4548 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4549 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4550 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
4551 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
4552 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
4553 <blg|pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
4554 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4555 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
4556 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
4557 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4558 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
4559 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
4560 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
4561 <pmn> \textless = {100, }, \textgreater = { ,100},
4562 <pmn> \textvisiblespace = {100,100} % not in LY1
4563 }
4564

```

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

```

4565 <*cmr>
4566 \SetProtrusion
4567 [ name = lmr-T1,
4568   load = cmr-T1 ]
4569 { encoding = {T1,LY1},
4570   family = lmr }
4571 {
4572   \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4573 }
4574
4575 </cmr>

```

Settings for the QX encoding (generic and Times). It also includes some glyphs otherwise in TS1.

```

4576 <*m-t|ptm>
4577 \SetProtrusion
4578 <m-t> [ name = QX-default,
4579 <ptm> [ name = ptm-QX,
4580 <m-t>   load = default ]
4581 <ptm>   load = ptm-default ]
4582 <m-t> { encoding = QX }
4583 <ptm> { encoding = QX,

```

```

4584 <ptm>    family = {ptm,ptmx,ptmj} }
4585 {
4586 <ptm>    * = {200,200},
4587    {=} = {100,100},
4588    \textunderscore = {100,100},
4589    \textbackslash = {100,200},
4590    \quotedblbase = {400,400},
4591 <m-t>    \guillemotleft = {200,200}, \guillemotright = {200,200},
4592 <ptm>    \guillemotleft = {300,300}, \guillemotright = {200,400},
4593    \textexclamdown = {100, }, \textquestiondown = {100, },
4594 <m-t>    \textbraceleft = {400,200}, \textbraceright = {200,400},
4595 <ptm>    \textbraceleft = {200,200}, \textbraceright = {200,300},
4596    \textless = {200,100}, \textgreater = {100,200},
4597    \textminus = {200,200}, \textdegree = {300,300},
4598 <m-t>    \copyright = {100,100}, \textregistered = {100,100}
4599 <ptm>    \copyright = {100,150}, \textregistered = {100,150},
4600 <ptm>    \textxgeq = { ,100}, \textxleq = {100, },
4601 <ptm>    \textalpha = { , 50}, \textDelta = { 70, 70},
4602 <ptm>    \textpi = { 50, 80}, \textSigma = { , 70},
4603 <ptm>    \textmu = { , 80}, \texteuro = { 50, 50},
4604 <ptm>    \textellipsis = {150,200}, \textasciitilde = { 80, 80},
4605 <ptm>    \textapprox = { 50, 50}, \textinfty = {100,100},
4606 <ptm>    \textdagger = {150,150}, \textdaggerdbl = {100,100},
4607 <ptm>    \textdiv = { 50,150}, \textsection = { 80, 80},
4608 <ptm>    \texttimes = {100,150}, \textpm = { 50, 80},
4609 <ptm>    \textbullet = {150,150}, \textperiodcentered = {300,300},
4610 <ptm>    \textquotesingle = {500,500}, \textquotedbl = {300,300},
4611 <ptm>    \textperthousand = { ,50}
4612 }
4613
4614 </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.

```

4615 <*cmr|bch>
4616 \SetProtrusion
4617 <cmr> [ name = cmr-T5,
4618 <cmr>    load = cmr-default ]
4619 <bch> [ name = bch-T5,
4620 <bch>    load = bch-default ]
4621 { encoding = T5,
4622 <cmr>    family = cmr }
4623 <bch>    family = bch }
4624 {
4625 <bch>    _ = {100,100},
4626 <bch>    \textbackslash = {150,200},
4627 <cmr>    \textbackslash = {200,300},
4628 <cmr>    \textquotedblleft = {200,600},
4629 <cmr>    \textquotedbl = {300,300},
4630 <bch>    \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4631 <cmr>    \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4632 <bch>    \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4633 <cmr>    \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4634 <bch>    \guillemotleft = {200,200}, \guillemotright = {150,300},
4635 <cmr>    \guillemotleft = {300,200}, \guillemotright = {100,400},
4636 <bch>    \textbraceleft = {200, }, \textbraceright = { ,300},
4637 <cmr>    \textbraceleft = {400,200}, \textbraceright = {200,400},
4638    \textless = {200,100}, \textgreater = {100,200}
4639 }
4640
4641 </cmr|bch>
4642 <*pmn>

```

```

4643 \SetProtrusion
4644 [ name      = pmnx-OT1,
4645   load      = pmnj-default ]
4646 { encoding = OT1,
4647   family   = pmnx }
4648 {
4649   1 = {230,180}
4650 }
4651
4652 \SetProtrusion
4653 [ name      = pmnx-T1,
4654   load      = pmnj-T1 ]
4655 { encoding = {T1,LY1},
4656   family   = pmnx }
4657 {
4658   1 = {230,180}
4659 }
4660
4661 /pmn

```

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

```

4662 (*ptm)
4663 \SetProtrusion
4664 [ name      = ptm-LY1,
4665   load      = ptm-T1 ]
4666 { encoding = LY1,
4667   family   = {ptm,ptmx,ptmj} }
4668 {
4669   -                      = {100,100},
4670   \texttrademark         = {100,100},
4671   \textregistered        = {100,100},
4672   \textcopyright         = {100,100},
4673   \textdegree            = {300,300},
4674   \textminus             = {200,200},
4675   \textellipsis          = {150,200},
4676   \texteuro              = {  ,  }, % ?
4677   \textcent              = {100,100},
4678   \textquotesingle       = {500,500},
4679   \textflorin            = { 50, 70},
4680   \textdagger            = {150,150},
4681   \textdaggerdbl         = {100,100},
4682   \textperthousand       = {  , 50},
4683   \textbullet            = {150,150},
4684   \textonesuperior       = {100,100},
4685   \texttwosuperior       = { 50, 50},
4686   \textthreesuperior     = { 50, 50},
4687   \textperiodcentered    = {300,300},
4688   \textplusminus         = { 50, 80},
4689   \textmultiply          = {100,100},
4690   \textdivide            = { 50,150}

```

Remaining slots in the source file.

```

4691   }
4692
4693 /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.

```

4694 \SetProtrusion
4695 <m-t> [ name      = OT1-it  ]
4696 <bch> [ name      = bch-it  ]
4697 <blg> [ name      = blg-it,
4698 <blg> [ load      = blg-default ]
4699 <cmr> [ name      = cmr-it  ]
4700 <pad> [ name      = pad-it  ]
4701 <pmn> [ name      = pmnj-it ]
4702 <ppl> [ name      = ppl-it  ]
4703 <ptm> [ name      = ptm-it  ]
4704 <ugm> [ name      = ugm-it  ]
4705 <m-t|bch|blg|pad|pmn|ugm> { encoding = OT1,
4706 <ppl|ptm> { encoding = {OT1,OT4},
4707 <bch>      family   = bch,
4708 <blg>      family   = blg,
4709 <pad>      family   = {pad,padx,padj},
4710 <pmn>      family   = pmnj,
4711 <ppl>      family   = {ppl,pplx,pplj},
4712 <ptm>      family   = {ptm,ptmx,ptmj},
4713 <ugm>      family   = ugm,
4714 <m-t|bch|pad|pmn|ppl|ptm>      shape   = {it,sl} }
4715 <blg|ugm>      shape   = it }
4716 <cmr> { }
4717 {
4718 <cmr|ptm>      A = {100,50},
4719 <pad|pmn>      A = {50, },
4720 <ugm>          A = { ,150},
4721 <ppl>          A = {50,50},
4722 <ptm>          \AE = {100, },
4723 <pad|ppl>      \AE = {50, },
4724 <pmn>          \AE = { , -50},
4725 <cmr|pad|ppl|ptm>      B = {50, },
4726 <pmn>          B = {20,-50},
4727 <bch|ppl|ptm|ugm>      C = {50, },
4728 <cmr|pad>      C = {100, },
4729 <pmn>          C = {50,-50},
4730 <cmr|pad|ppl|ptm>      D = {50,50},
4731 <pmn>          D = {20, },
4732 <cmr|pad|ppl|ptm>      E = {50, },
4733 <pmn>          E = {20,-50},
4734 <cmr|pad|ptm>        F = {100, },
4735 <pmn>          F = {10, },
4736 <ppl>          F = {50, },
4737 <bch|ppl|ptm|ugm>      G = {50, },
4738 <cmr|pad>        G = {100, },
4739 <pmn>          G = {50,-50},
4740 <cmr|pad|ppl|ptm>      H = {50, },
4741 <cmr|pad|ptm>        I = {50, },
4742 <pmn>          I = {20,-50},
4743 <cmr|ptm>        J = {100, },
4744 <pad>          J = {50, },
4745 <pmn>          J = {20, },
4746 <cmr|pad|ppl|ptm>      K = {50, },
4747 <pmn>          K = {20, },
4748 <cmr|pad|ppl|ptm>      L = {50, },
4749 <pmn>          L = {20,50},
4750 <ugm>          L = { ,100},
4751 <cmr|ptm>        M = {50, },
4752 <pmn>          M = { , -30},
4753 <cmr|ptm>        N = {50, },
4754 <pmn>          N = { , -30},

```

```

4755 <bch|pmn|ppl|ptm>    0 = {50, },
4756 <cmr|pad>           0 = {100, },
4757 <ugm>               0 = {70,50},
4758 <pmn|ppl|ptm>       \OE = {50, },
4759 <pad>               \OE = {100, },
4760 <cmr|pad|ppl|ptm>   P = {50, },
4761 <pmn>               P = {20,-50},
4762 <bch|pmn|ppl|ptm>    Q = {50, },
4763 <cmr|pad>           Q = {100, },
4764 <ugm>               Q = {70,50},
4765 <cmr|pad|ppl|ptm>   R = {50, },
4766 <pmn>               R = {20, },
4767 <bch|cmr|pad|ppl|ptm> S = {50, },
4768 <pmn>               S = {20,-30},
4769 <bch|cmr|pad|ppl|ptm> $ = {50, },
4770 <pmn>               $ = {20,-30},
4771 <bch|pmn|ugm>       T = {70, },
4772 <cmr|pad|ppl|ptm>   T = {100, },
4773 <cmr|pad|ppl|ptm>   U = {50, },
4774 <pmn>               U = {50,-50},
4775 <cmr|pad|pmn|ugm>   V = {100, },
4776 <ppl|ptm>           V = {100,50},
4777 <cmr|pad|pmn|ugm>   W = {100, },
4778 <ppl>               W = {50, },
4779 <ptm>               W = {100,50},
4780 <cmr|ppl|ptm>       X = {50, },
4781 <cmr|ptm>           Y = {100, },
4782 <pmn>               Y = {50, },
4783 <ppl>               Y = {100,50},
4784 <pmn>               Z = { , -50},
4785 <pmn>               d = { , -50},
4786 <pad|pmn>           f = { , -100},
4787 <pmn>               i = { , -30},
4788 <pmn>               j = { , -30},
4789 <pmn>               l = { , -100},
4790 <bch>               o = {50,50},
4791 <bch>               p = { , 50},
4792 <pmn>               p = {-50, },
4793 <bch>               q = {50, },
4794 <pmn>               r = { , 50},
4795 <bch>               t = { , 50},
4796 <pmn|ugm>           v = {50, },
4797 <bch>               w = { , 50},
4798 <pmn|ugm>           w = {50, },
4799 <bch>               y = { , 50},
4800 <cmr>               0 = {100, },
4801 <bch|ptm>           1 = {150,100},
4802 <cmr>               1 = {200,50},
4803 <pad>               1 = {150, },
4804 <pmn>               1 = {50, },
4805 <ppl>               1 = {100, },
4806 <ugm>               1 = {150,150},
4807 <cmr>               2 = {100,-100},
4808 <pad|ppl|ptm>       2 = {50, },
4809 <pmn>               2 = {-50, },
4810 <bch>               3 = {50, },
4811 <cmr>               3 = {100,-100},
4812 <pmn>               3 = {-100, },
4813 <ptm>               3 = {100,50},
4814 <bch>               4 = {100, },
4815 <cmr|pad>           4 = {150, },
4816 <ppl|ptm>           4 = {50, },
4817 <cmr>               5 = {100, },

```

```

4818 <ptm>      5 = {50, },
4819 <bch>      6 = {50, },
4820 <cmr>      6 = {100, },
4821 <bch|pad|ptm> 7 = {100, },
4822 <cmr>      7 = {200,-150},
4823 <pmn>      7 = {20, },
4824 <ppl>      7 = {50, },
4825 <cmr>      8 = {50,-50},
4826 <cmr>      9 = {100,-100},
4827 <m-t|cmr|pad|pmn|ppl> . = { ,500},
4828 <blg>      . = {400,600},
4829 <bch|ptm|ugm> . = { ,700},
4830 <blg>      {,}= {300,500},
4831 <m-t|cmr|pad|pmn|ppl> {,}= { ,500},
4832 <bch|ugm>   {,}= { ,600},
4833 <ptm>      {,}= { ,700},
4834 <m-t|cmr|pad|ppl> : = { ,300},
4835 <bch|ugm>   : = { ,400},
4836 <pmn>      : = { ,200},
4837 <ptm>      : = { ,500},
4838 <m-t|cmr|pad|ppl> ; = { ,300},
4839 <bch|ugm>   ; = { ,400},
4840 <pmn>      ; = { ,200},
4841 <ptm>      ; = { ,500},
4842 <ptm>      ! = { ,100},
4843 <bch>      ? = { ,200},
4844 <ptm>      ? = { ,100},
4845 <ppl>      ? = { ,300},
4846 <pmn>      " = {400,200},
4847 <m-t|pad|pmn|ppl|ptm> & = {50,50},
4848 <bch>      & = { ,80},
4849 <cmr>      & = {100,50},
4850 <ugm>      & = {50,100},
4851 <m-t|cmr|pad|pmn> \% = {100, },
4852 <bch>      \% = {50,50},
4853 <ppl|ptm>   \% = {100,100},
4854 <ugm>      \% = {100,50},
4855 <m-t|pmn|ppl> * = {200,200},
4856 <bch>      * = {300,200},
4857 <cmr>      * = {400,100},
4858 <pad>      * = {500,100},
4859 <ptm|ugm>  * = {400,200},
4860 <m-t|cmr|pmn|ppl> + = {150,200},
4861 <bch|ugm>  + = {250,250},
4862 <pad|ptm>  + = {250,200},
4863 <m-t|pad|pmn|ppl> @ = {50,50},
4864 <bch>      @ = {80,50},
4865 <cmr>      @ = {200,50},
4866 <ptm>      @ = {150,150},
4867 <m-t|bch|ugm> ~ = {150,150},
4868 <cmr|pad|pmn|ppl|ptm> ~ = {200,150},
4869 <ugm>      {=} = {200,200},
4870 <!blg>     ( = {200, }, ) = { ,200},
4871 <m-t|cmr|pad|ppl|ptm|ugm> / = {100,200},
4872 <bch>      / = { ,150},
4873 <pmn>      / = {100,150},
4874 <m-t>      - = {300,300},
4875 <bch|pad>  - = {300,400},
4876 <pmn>      - = {200,300},
4877 <cmr>      - = {500,300},
4878 <ppl>      - = {300,500},
4879 <ptm>      - = {500,500},
4880 <ugm>      - = {400,700},

```



```

4881 <blg>      _ = {0,300},
4882 <m-t|pmn>   \textendash      = {200,200}, \textemdash      = {150,150},
4883 <bch>       \textendash      = {200,300}, \textemdash      = {150,200},
4884 <cmr>       \textendash      = {500,300}, \textemdash      = {400,200},
4885 <pad|ppl|ptm|ugm> \textendash      = {300,300}, \textemdash      = {200,200},
4886 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
4887 <blg>       \textquoteleft = {400,400}, \textquoteright = {400,400},
4888 <cmr|pad>    \textquoteleft = {800,200}, \textquoteright = {800,200},
4889 <ppl>        \textquoteleft = {700,400}, \textquoteright = {700,400},
4890 <ptm>        \textquoteleft = {800,500}, \textquoteright = {800,500},
4891 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
4892 <blg>       \textquotedblright = {300,300}
4893 <cmr>       \textquotedblleft = {700,100}, \textquotedblright = {500,300}
4894 <pad>       \textquotedblleft = {700,200}, \textquotedblright = {700,200}
4895 <ppl>       \textquotedblleft = {500,300}, \textquotedblright = {500,300}
4896 <ptm>       \textquotedblleft = {700,400}, \textquotedblright = {700,400}
4897 <ugm>       \textquotedblleft = {600,200}, \textquotedblright = {600,200}
4898 }
4899
4900 <*cmr>
4901 \SetProtrusion
4902 [ name      = cmr-it-OT1,
4903   load      = cmr-it  ]
4904 { encoding = {OT1,OT4},
4905   family   = cmr,
4906   shape     = it      }
4907 {
4908   \AE = {100,  },
4909   \OE = {100,  },
4910   "00 = {200,150}, % \Gamma
4911   "01 = {150,100}, % \Delta
4912   "02 = {150, 50}, % \Theta
4913   "03 = {150, 50}, % \Lambda
4914   "04 = {100,100}, % \Xi
4915   "05 = {100,100}, % \Pi
4916   "06 = {100, 50}, % \Sigma
4917   "07 = {200,150}, % \Upsilon
4918   "08 = {150, 50}, % \Phi
4919   "09 = {150,100}, % \Psi
4920   "0A = { 50, 50} % \Omega
4921 }
4922
4923 </cmr>
4924 \SetProtrusion
4925 <m-t> [ name      = T1-it-default,
4926 <bch> [ name      = bch-it-T1,
4927 <blg> [ name      = blg-it-T1,
4928 <cmr> [ name      = cmr-it-T1,
4929 <pad> [ name      = pad-it-T1,
4930 <pmn> [ name      = pmnj-it-T1,
4931 <ppl> [ name      = ppl-it-T1,
4932 <ptm> [ name      = ptm-it-T1,
4933 <ugm> [ name      = ugm-it-T1,
4934 <m-t> load      = OT1-it  ]
4935 <bch> load      = bch-it  ]
4936 <blg> load      = blg-T1  ]
4937 <cmr> load      = cmr-it  ]
4938 <pmn> load      = pmnj-it ]
4939 <pad> load      = pad-it  ]
4940 <ppl> load      = ppl-it  ]
4941 <ptm> load      = ptm-it  ]
4942 <ugm> load      = ugm-it  ]
4943 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},

```

```

4944 <blg|ptm|ugm> { encoding = {Tl},
4945 <bch> family = bch,
4946 <blg> family = blg,
4947 <cmr> family = cmr,
4948 <pmn> family = pmnj,
4949 <pad> family = {pad,padx,padj},
4950 <ppl> family = {ppl,pplx,pplj},
4951 <ptm> family = {ptm,ptmx,ptmj},
4952 <ugm> family = ugm,
4953 <m-t|bch|pad|pmn|ppl|ptm> shape = {it,sl} }
4954 <blg|cmr|ugm> shape = it }
4955 {
4956 <m-t|bch|pmn> _ = { ,100},
4957 <blg> _ = {0,300},
4958 <cmr|ugm> _ = {100,200},
4959 <pad|ppl|ptm> _ = {100,100},
4960 <blg> . = {400,600},
4961 <blg> {,}= {300,500},
4962 <cmr> \AE = {100, },
4963 <bch> \OE = { 50, },
4964 <cmr> \OE = {100, },
4965 <pmn> 031 = { , -100}, % ffl
4966 <cmr|ptm> 156 = {100, }, % IJ
4967 <pad> 156 = {50, }, % IJ
4968 <pmn> 156 = {20, }, % IJ
4969 <pmn> 188 = { , -30}, % ij
4970 <pmn> \v t = { ,100},
4971 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
4972 <cmr|ugm> \textbackslash = {300,300},
4973 <bch> \textbackslashleft = {150,150},
4974 <pmn> \textbackslash = {100,150},
4975 <ugm> \textbar = {200,200},
4976 <cmr> \textquotedblleft = {500,300},
4977 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
4978 <blg> \textquotedbl = {300,300}, \textquotedblleft = {300,300},
4979 <blg> \textquotedblright = {300,300}, \quotedblbase = {200,600},
4980 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
4981 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4982 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4983 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
4984 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
4985 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4986 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4987 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4988 <pad> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
4989 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4990 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
4991 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
4992 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4993 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
4994 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
4995 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4996 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
4997 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4998 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
4999 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
5000 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
5001 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
5002 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
5003 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
5004 <pmn> \textvisiblespace = {100,100}
5005 }
5006

```

```

5007 <*m-t|ptm>
5008 \SetProtrusion
5009 <m-t> [ name = QX-it-default,
5010 <ptm> [ name = ptm-it-QX,
5011 <m-t> load = OT1-it ]
5012 <ptm> load = ptm-it ]
5013 { encoding = {QX},
5014 <ptm> family = {ptm,ptmx,ptmj},
5015 shape = {it,sl} }
5016 {
5017 <ptm> 009 = { , 50}, % fk
5018 {=} = {100,100},
5019 <m-t> \textunderscore = {100,100},
5020 <ptm> \textunderscore = {100,150},
5021 \textbackslash = {100,200},
5022 \quotedblbase = {300,400},
5023 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
5024 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
5025 \textexclamdown = {200, }, \textquestiondown = {200, },
5026 \textbraceleft = {200,100}, \textbraceright = {200,200},
5027 \textless = {100,100}, \textgreater = {100,100},
5028 \textminus = {200,200}, \textdegree = {300,150},
5029 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5030 <ptm> \textregistered = {100,150}, \copyright = {100,150},
5031 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
5032 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
5033 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
5034 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
5035 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
5036 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
5037 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5038 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
5039 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
5040 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
5041 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5042 <ptm> \textperthousand = { ,50}
5043 }
5044
5045 </m-t|ptm>
5046 <*cmr|bch>
5047 \SetProtrusion
5048 <cmr> [ name = cmr-it-T5,
5049 <cmr> load = cmr-it ]
5050 <bch> [ name = bch-it-T5,
5051 <bch> load = bch-it ]
5052 { encoding = T5,
5053 <bch> family = bch,
5054 <cmr> family = cmr,
5055 shape = it }
5056 {
5057 <bch> _ = { ,100},
5058 <cmr> _ = {100,200},
5059 <bch> \textbackslash = {150,150},
5060 <cmr> \textbackslash = {300,300},
5061 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
5062 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5063 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5064 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5065 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
5066 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
5067 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
5068 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
5069 <bch> \textless = {100, }, \textgreater = { ,100}

```

```

5070 <cmr> \textless = {300,100}, \textgreater = {200,100}
5071 }
5072
5073 </cmr|bch>

```

Slanted is very similar to italic.

```

5074 <*cmr>
5075 \SetProtrusion
5076 [ name = cmr-sl,
5077   load = cmr-it-OT1 ]
5078 { encoding = {OT1,OT4},
5079   family = cmr,
5080   shape = sl }
5081 {
5082   L = { ,50},
5083   f = { ,-50},
5084   - = {300, },
5085   \textendash = {400, }, \textemdash = {300, }
5086 }
5087
5088 \SetProtrusion
5089 [ name = cmr-sl-T1,
5090   load = cmr-it-T1 ]
5091 { encoding = {T1,LY1},
5092   family = cmr,
5093   shape = sl }
5094 {
5095   L = { ,50},
5096   f = { ,-50},
5097   - = {300, },
5098   \textendash = {400, }, \textemdash = {300, }
5099 }
5100
5101 \SetProtrusion
5102 [ name = cmr-sl-T5,
5103   load = cmr-it-T5 ]
5104 { encoding = T5,
5105   family = cmr,
5106   shape = sl }
5107 {
5108   L = { ,50},
5109   f = { ,-50},
5110   - = {300, },
5111   \textendash = {400, }, \textemdash = {300, }
5112 }
5113
5114 \SetProtrusion
5115 [ name = lmr-it-T1,
5116   load = cmr-it-T1 ]
5117 { encoding = {T1,LY1},
5118   family = lmr,
5119   shape = {it,sl} }
5120 {
5121   \textquotedblleft = { ,200}, \textquotedblright = { ,200},
5122   \quotesinglbase = { ,400}, \quotedblbase = { ,500}
5123 }
5124

```

Oldstyle numerals are slightly different.

```

5125 \SetProtrusion
5126 [ name = cmr(oldstyle)-it,
5127   load = cmr-it-T1 ]
5128 { encoding = T1,

```

```

5129     family    = {hfor,cmor},
5130     shape      = {it,sl} }
5131   {
5132     1 = {250, 50},
5133     2 = {150,-100},
5134     3 = {100,-50},
5135     4 = {150,150},
5136     6 = {200,  },
5137     7 = {200, 50},
5138     8 = {150,-50},
5139     9 = {100, 50}
5140   }
5141
5142 /cmr
5143 *pmn
5144 \SetProtrusion
5145   [ name      = pmnx-it,
5146     load      = pmnj-it ]
5147   { encoding = OT1,
5148     family    = pmnx,
5149     shape      = {it,sl} }
5150   {
5151     1 = {100,150}
5152   }
5153
5154 \SetProtrusion
5155   [ name      = pmnx-it-T1,
5156     load      = pmnj-it-T1 ]
5157   { encoding = {T1,LY1},
5158     family    = pmnx,
5159     shape      = {it,sl} }
5160   {
5161     1 = {100,150}
5162   }
5163
5164 /pmn
5165 *ptm
5166 \SetProtrusion
5167   [ name      = ptm-it-LY1,
5168     load      = ptm-it-T1 ]
5169   { encoding = {LY1},
5170     family    = {ptm,ptmx,ptmj},
5171     shape      = {it,sl} }
5172   {
5173     -                      = {100,100},
5174     \texttrademark         = {100,100},
5175     \textregistered        = {100,100},
5176     \textcopyright         = {100,100},
5177     \textdegree            = {300,100},
5178     \textminus             = {200,200},
5179     \textellipsis          = {100,200},
5180     \texteuro              = {  ,  },
5181     \textcent              = {100,100},
5182     \textquotesingle       = {500,  },
5183     \textflorin            = {100, 70},
5184     \textdagger            = {150,150},
5185     \textdaggerdbl         = {100,100},
5186     \textbullet            = {150,150},
5187     \textonesuperior       = {150,100},
5188     \texttwosuperior       = {150, 50},
5189     \textthreesuperior     = {150, 50},
5190     \textparagraph         = {100,  },
5191     \textperiodcentered    = {500,300},

```

```

5192 \textonequarter      = { 50,  },
5193 \textonehalf         = { 50,  },
5194 \textplusminus        = {100,100},
5195 \textmultiply         = {150,150},
5196 \textdivide          = {150,150}
5197 }
5198
5199 </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).

```

5200 <*(blg|ugm)>
5201 \SetProtrusion
5202 <m-t> [ name      = OT1-sc,
5203 <bch> [ name      = bch-sc,
5204 <cmr> [ name      = cmr-sc-OT1,
5205 <pad> [ name      = pad-sc,
5206 <pmn> [ name      = pmnj-sc,
5207 <ppl> [ name      = ppl-sc,
5208 <ptm> [ name      = ptm-sc,
5209 <m-t> load      = default ]
5210 <bch> load      = bch-default ]
5211 <cmr> load      = cmr-OT1 ]
5212 <pad> load      = pad-default ]
5213 <pmn> load      = pmnj-default ]
5214 <ppl> load      = ppl-default ]
5215 <ptm> load      = ptm-default ]
5216 <m-t|bch|pad|pmn> { encoding = OT1,
5217 <cmr|ppl|ptm> { encoding = {OT1,OT4},
5218 <bch> family     = bch,
5219 <cmr> family     = cmr,
5220 <pad> family     = {pad,padx,padj},
5221 <pmn> family     = pmnj,
5222 <ppl> family     = {ppl,pplx,pplj},
5223 <ptm> family     = {ptm,ptmx,ptmj},
5224 shape          = sc }
5225 {
5226 a = {50,50},
5227 <cmr|pad|ppl|ptm> \ae = {50,  },
5228 <bch|pmn> c = {50,  },
5229 <bch|pad|pmn> d = { ,50},
5230 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
5231 <bch|pad|pmn> g = {50,  },
5232 <m-t|cmr|pad|pmn|ppl|ptm> j = {50,  },
5233 <bch> j = {100,  },
5234 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
5235 <ptm> l = { ,80},
5236 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
5237 <ptm> 013 = { ,80}, % fl
5238 <bch|pad|pmn> o = {50,50},
5239 <pad|pmn> \oe = {50,  },
5240 <ppl> p = { 0, 0},
5241 <bch|pad|pmn> q = {50,70},
5242 <ppl> q = { 0,  },
5243 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5244 t = {50,50},
5245 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5246 <ptm> y = {80,80}

```



```

5310     t = {50,50},
5311     y = {50,50}
5312 }
5313
5314 </m-t>
5315 <*cmr|bch>
5316 \SetProtrusion
5317 <bch> [ name      = bch-sc-T5,
5318 <bch>   load      = bch-T5 ]
5319 <cmr> [ name      = cmr-sc-T5,
5320 <cmr>   load      = cmr-T5 ]
5321 { encoding = T5,
5322 <bch>   family   = bch,
5323 <cmr>   family   = cmr,
5324   shape    = sc }
5325 {
5326   a = {50,50},
5327 <bch>   c = {50, },
5328 <bch>   d = {  ,50},
5329   f = {  ,50},
5330 <bch>   g = {50, },
5331 <bch>   j = {100, },
5332 <cmr>   j = {50, },
5333   l = {  ,50},
5334 <bch>   o = {50,50},
5335 <bch>   q = { 0, },
5336 <cmr>   r = {  , 0},
5337   t = {50,50},
5338   y = {50,50}
5339 }
5340
5341 </cmr|bch>
5342 <*pmn>
5343 \SetProtrusion
5344 [ name      = pmnx-sc,
5345   load      = pmnj-sc ]
5346 { encoding = OT1,
5347   family   = pmnx,
5348   shape    = sc }
5349 {
5350   l = {230,180}
5351 }
5352
5353 \SetProtrusion
5354 [ name      = pmnx-sc-T1,
5355   load      = pmnj-sc-T1 ]
5356 { encoding = {T1,LY1},
5357   family   = pmnx,
5358   shape    = sc }
5359 {
5360   l = {230,180}
5361 }
5362

```

#### 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`.

```

5363 \SetProtrusion
5364 [ name      = pmnj-scit,
5365   load      = pmnj-it ]

```



```

5366 { encoding = OT1,
5367     family   = pmnj,
5368     shape    = {scit,si} }
5369 {
5370     a = {50, },
5371     \ae = { , -50},
5372     b = {20, -50},
5373     c = {50, -50},
5374     d = {20, 0},
5375     e = {20, -50},
5376     f = {10, 0},
5377     012 = {10, -50}, % fi
5378     013 = {10, -50}, % fl
5379     014 = {10, -50}, % ffi
5380     015 = {10, -50}, % ffl
5381     g = {50, -50},
5382     i = {20, -50},
5383     j = {20, 0},
5384     k = {20, },
5385     l = {20, 50},
5386     m = { , -30},
5387     n = { , -30},
5388     o = {50, },
5389     \oe = {50, -50},
5390     p = {20, -50},
5391     q = {50, },
5392     r = {20, 0},
5393     s = {20, -30},
5394     t = {70, },
5395     u = {50, -50},
5396     v = {100, },
5397     w = {100, },
5398     y = {50, },
5399     z = { , -50}
5400 }
5401
5402 \SetProtrusion
5403 [ name   = pmnj-scit-T1,
5404   load   = pmnj-it-T1 ]
5405 { encoding = {T1,LY1},
5406   family   = pmnj,
5407   shape    = {scit,si} }
5408 {
5409     a = {50, },
5410     \ae = { , -50},
5411     b = {20, -50},
5412     c = {50, -50},
5413     d = {20, 0},
5414     e = {20, -50},
5415     f = {10, 0},
5416     028 = {10, -50}, % fi
5417     029 = {10, -50}, % fl
5418     030 = {10, -50}, % ffi
5419     031 = {10, -50}, % ffl
5420     g = {50, -50},
5421     i = {20, -50},
5422     188 = {20, 0}, % ij
5423     j = {20, 0},
5424     k = {20, },
5425     l = {20, 50},
5426     m = { , -30},
5427     n = { , -30},
5428     o = {50, },

```

```

5429 \oe = {50,-50},
5430 p = {20,-50},
5431 q = {50, },
5432 r = {20, 0},
5433 s = {20,-30},
5434 t = {70, },
5435 u = {50,-50},
5436 v = {100, },
5437 w = {100, },
5438 y = {50, },
5439 z = { , -50}
5440 }
5441
5442 \SetProtrusion
5443 [ name = pmnx-scit,
5444 load = pmnj-scit ]
5445 { encoding = OT1,
5446 family = pmnx,
5447 shape = {scit,si} }
5448 {
5449 1 = {100,150}
5450 }
5451
5452 \SetProtrusion
5453 [ name = pmnx-scit-T1,
5454 load = pmnj-scit-T1 ]
5455 { encoding = {T1,LY1},
5456 family = pmnx,
5457 shape = {scit,si} }
5458 {
5459 1 = {100,150}
5460 }
5461
5462 </pmn>

```

### 15.8.5 Text Companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

5463 \SetProtrusion
5464 <m-t> [ name = textcomp ]
5465 <bch> [ name = bch-textcomp ]
5466 <blg> [ name = blg-textcomp ]
5467 <cmr> [ name = cmr-textcomp ]
5468 <pad> [ name = pad-textcomp ]
5469 <pmn> [ name = pmn-textcomp ]
5470 <ppl> [ name = ppl-textcomp ]
5471 <ptm> [ name = ptm-textcomp ]
5472 <ugm> [ name = ugm-textcomp ]
5473 <m-t> { encoding = TS1 }
5474 <!m-t> { encoding = TS1,
5475 <bch> family = bch }
5476 <blg> family = blg }
5477 <cmr> family = cmr }
5478 <pad> family = {pad,padx,padj} }
5479 <pmn> family = {pmnx,pmnj} }
5480 <ppl> family = {ppl,pplx,pplj} }
5481 <ptm> family = {ptm,ptmx,ptmj} }
5482 <ugm> family = ugm }
5483 {
5484 <blg> \textquotestraightbase = {400,500},

```

```

5485 <cmr> \textquotestraightbase = {300,300},
5486 <pad|pmn> \textquotestraightbase = {400,400},
5487 <blg> \textquotestraightdblbase = {300,400},
5488 <cmr|pmn> \textquotestraightdblbase = {300,300},
5489 <pad> \textquotestraightdblbase = {400,400},
5490 <bch|cmr|pad|pmn|ugm> \texttwelveudash = {200,200},
5491 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
5492 <ugm> \textthreequartersemdash = {200,200},
5493 <blg> \textquotesingle = {500,600},
5494 <cmr|pmn> \textquotesingle = {300,400},
5495 <pad> \textquotesingle = {400,500},
5496 <ptm> \textquotesingle = {500,500},
5497 <ugm> \textquotesingle = {300,500},
5498 <bch|cmr|pmn> \textasteriskcentered = {200,300},
5499 <blg> \textasteriskcentered = {150,200},
5500 <pad> \textasteriskcentered = {300,300},
5501 <ugm> \textasteriskcentered = {100,200},
5502 <pmn> \textfractionsolidus = {-200,-200},
5503 <cmr> \textoneoldstyle = {100,100},
5504 <pmn> \textoneoldstyle = { , 50},
5505 <cmr> \textthreeoldstyle = { , 50},
5506 <pad|pmn> \textthreeoldstyle = { 50, },
5507 <cmr> \textfouroldstyle = { 50, 50},
5508 <pad|pmn> \textfouroldstyle = { 50, },
5509 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
5510 <cmr> \textlangle = {400, },
5511 <cmr> \textrangle = { ,400},
5512 <m-t|bch|pmn|ptm> \textminus = {200,200},
5513 <cmr|pad|ppl> \textminus = {300,300},
5514 <blg|ugm> \textminus = {250,300},
5515 <bch|pad|pmn> \textlbrackdbl = {100, },
5516 <blg> \textlbrackdbl = {200, },
5517 <bch|pad|pmn> \textrbrackdbl = { ,100},
5518 <blg> \textrbrackdbl = { ,200},
5519 <pmn> \textasciigrave = {200,500},
5520 <bch|blg|cmr|pad|pmn> \texttildelow = {200,250},
5521 <pmn> \textasciibreve = {300,400},
5522 <pmn> \textasciicaron = {300,400},
5523 <pmn> \textacutedbl = {200,300},
5524 <pmn> \textgravedbl = {150,300},
5525 <bch|pmn|ugm> \textdagger = { 80, 80},
5526 <blg> \textdagger = {200,200},
5527 <cmr|pad> \textdagger = {100,100},
5528 <ptm> \textdagger = {150,150},
5529 <blg> \textdaggerdbl = {150,150},
5530 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
5531 <ptm> \textdaggerdbl = {100,100},
5532 <bch> \textbardbl = {100,100},
5533 <blg|ugm> \textbardbl = {150,150},
5534 <bch> \textbullet = {200,200},
5535 <blg> \textbullet = {400,500},
5536 <cmr|pad|pmn> \textbullet = { ,100},
5537 <ptm> \textbullet = {150,150},
5538 <ugm> \textbullet = { 50,100},
5539 <bch|cmr|pmn> \textcelsius = { 50, },
5540 <pad> \textcelsius = { 80, },
5541 <bch> \textflorin = { 50, 50},
5542 <blg> \textflorin = {100,100},
5543 <pad|ugm> \textflorin = { ,100},
5544 <pmn> \textflorin = { 50,100},
5545 <ptm> \textflorin = { 50, 70},
5546 <cmr> \textcolonmonetary = { , 50},
5547 <pad|pmn> \textcolonmonetary = { 50, },

```

```

5548 <pmn> \textinterrobang = { ,100},
5549 <pmn> \textinterrobangdown = {100, },
5550 <m-t|pad|ptm> \texttrademark = {100,100},
5551 <bch> \texttrademark = {150,150},
5552 <blg|cmr|ppl> \texttrademark = {200,200},
5553 <pmn> \texttrademark = { 50, 50},
5554 <ugm> \texttrademark = {100,150},
5555 <bch|ugm> \textcent = { 50, },
5556 <ptm> \textcent = {100,100},
5557 <bch> \textsterling = { 50, },
5558 <ugm> \textsterling = { , 50},
5559 <bch> \textbrokenbar = {200,200},
5560 <blg> \textbrokenbar = {250,250},
5561 <ugm> \textbrokenbar = {200,300},
5562 <pmn> \textasciidieresis = {300,400},
5563 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
5564 <pmn> \textcopyright = {100,150},
5565 <ppl> \textcopyright = {200,200},
5566 <bch|cmr|ugm> \textordfeminine = {100,200},
5567 <pad|pmn> \textordfeminine = {200,200},
5568 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
5569 <blg> \textlnot = {200,100},
5570 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
5571 <pmn> \textregistered = { 50,150},
5572 <ppl> \textregistered = {200,200},
5573 <pmn> \textasciimacron = {150,200},
5574 <m-t|ppl|ptm> \textdegree = {300,300},
5575 <bch> \textdegree = {150,200},
5576 <blg|ugm> \textdegree = {200,200},
5577 <cmr|pad> \textdegree = {400,400},
5578 <pmn> \textdegree = {150,400},
5579 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
5580 <blg> \textpm = {100,100},
5581 <ptm> \textpm = { 50, 80},
5582 <bch|blg|ugm> \texttwosuperior = {100,200},
5583 <cmr> \texttwosuperior = { 50,100},
5584 <pad|pmn> \texttwosuperior = {200,200},
5585 <ptm> \texttwosuperior = { 50, 50},
5586 <bch|blg|ugm> \textthreesuperior = {100,200},
5587 <cmr> \textthreesuperior = { 50,100},
5588 <pad|pmn> \textthreesuperior = {200,200},
5589 <ptm> \textthreesuperior = { 50, 50},
5590 <pmn> \textasciicute = {300,400},
5591 <bch|ugm> \textmu = { ,100},
5592 <bch|pad|pmn> \textparagraph = { ,100},
5593 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
5594 <blg> \textperiodcentered = {400,500},
5595 <ptm> \textperiodcentered = {300,300},
5596 <ugm> \textperiodcentered = {200,500},
5597 <bch|blg|ugm> \textonesuperior = {200,300},
5598 <cmr|pad|pmn> \textonesuperior = {200,200},
5599 <ptm> \textonesuperior = {100,100},
5600 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
5601 <blg|cmr> \textordmasculine = {100,200},
5602 <bch|cmr|pmn> \texteuro = {100, },
5603 <pad> \texteuro = { 50,100},
5604 <bch> \texttimes = {200,200},
5605 <blg|ptm> \texttimes = {100,100},
5606 <cmr> \texttimes = {150,250},
5607 <pad> \texttimes = {100,150},
5608 <pmn> \texttimes = { 70,100},
5609 <ugm> \texttimes = {200,300},
5610 <bch|pad|pmn> \textdiv = {150,200}

```

```

5611 <blg> \textdiv = {100,100}
5612 <cmr> \textdiv = {150,250}
5613 <ptm> \textdiv = { 50,100},
5614 <ugm> \textdiv = {200,300},
5615 <ptm> \textperthousand = { ,50}
5616 <ugm> \textsection = { ,100},
5617 <ugm> \textonehalf = { 50,100},
5618 <ugm> \textonequarter = { 50,100},
5619 <ugm> \textthreequarters = { 50,100},
5620 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

5621 }
5622
5623 <*cmr|pad|pmn|ugm>
5624 \SetProtrusion
5625 <cmr> [ name = cmr-textcomp-it ]
5626 <pad> [ name = pad-textcomp-it ]
5627 <pmn> [ name = pmn-textcomp-it ]
5628 <ugm> [ name = ugm-textcomp-it ]
5629 { encoding = TS1,
5630 <cmr> family = cmr,
5631 <pad> family = {pad,padx,padj},
5632 <pmn> family = {pmnx,pmnj},
5633 <ugm> family = ugm,
5634 <!ugm> shape = {it,sl} }
5635 <ugm> shape = it }
5636 {
5637 <cmr> \textquotestraightbase = {300,600},
5638 <pad|pmn> \textquotestraightbase = {400,400},
5639 <cmr> \textquotestraightdblbase = {300,600},
5640 <pad> \textquotestraightdblbase = {300,400},
5641 <pmn> \textquotestraightdblbase = {300,300},
5642 \texttwelveudash = {200,200},
5643 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
5644 <ugm> \textthreequartersemdash = {200,200},
5645 <cmr> \textquotesingle = {600,300},
5646 <pad> \textquotesingle = {800,100},
5647 <pmn> \textquotesingle = {300,200},
5648 <ugm> \textquotesingle = {500,500},
5649 <cmr> \textasteriskcentered = {300,200},
5650 <pad> \textasteriskcentered = {500,100},
5651 <pmn> \textasteriskcentered = {200,300},
5652 <ugm> \textasteriskcentered = {300,150},
5653 <pmn> \textfractionsolidus = {-200,-200},
5654 <cmr> \textoneoldstyle = {100, 50},
5655 <pad> \textoneoldstyle = {100, },
5656 <pmn> \textoneoldstyle = { 50, },
5657 <pad> \texttwooldstyle = { 50, },
5658 <pmn> \texttwooldstyle = {-50, },
5659 <cmr> \textthreeoldstyle = {100, 50},
5660 <pmn> \textthreeoldstyle = {-100, },
5661 <cmr> \textfouroldstyle = { 50, 50},
5662 <pad> \textfouroldstyle = { 50,100},
5663 <cmr> \textsevenoldstyle = { 50, 80},
5664 <pad> \textsevenoldstyle = { 50, },
5665 <pmn> \textsevenoldstyle = { 20, },
5666 <cmr> \textlangle = {400, },
5667 <cmr> \textrangle = { ,400},
5668 <cmr|pad> \textminus = {300,300},
5669 <pmn> \textminus = {200,200},
5670 <ugm> \textminus = {250,300},
5671 <pad|pmn> \textlbrackdbl = {100, },

```

```

5672 <pad|pmn> \textrbrackdbl = { ,100},
5673 <pmn> \textasciigrave = {300,300},
5674 <cmr|pad|pmn> \texttildelow = {200,250},
5675 <pmn> \textasciibreve = {300,300},
5676 <pmn> \textasciicaron = {300,300},
5677 <pmn> \textacutedbl = {200,300},
5678 <pmn> \textgravedbl = {150,300},
5679 <cmr> \textdagger = {100,100},
5680 <pad> \textdagger = {200,100},
5681 <pmn> \textdagger = { 80, 50},
5682 <ugm> \textdagger = { 80, 80},
5683 <cmr|pad> \textdaggerdbl = { 80, 80},
5684 <pmn> \textdaggerdbl = { 80, 50},
5685 <ugm> \textbardbl = {150,150},
5686 <cmr> \textbullet = {200,100},
5687 <pad> \textbullet = {300, },
5688 <pmn> \textbullet = { 30, 70},
5689 <ugm> \textbullet = { 50,100},
5690 <cmr> \textcelsius = {100, },
5691 <pad> \textcelsius = {200, },
5692 <pmn> \textcelsius = { 50,-50},
5693 <pad> \textflorin = {100, },
5694 <pmn> \textflorin = { 50,100},
5695 <ugm> \textflorin = { ,100},
5696 <cmr> \textcolonmonetary = {150, },
5697 <pad> \textcolonmonetary = {100, },
5698 <pmn> \textcolonmonetary = { 50,-50},
5699 <cmr|pad> \texttrademark = {200, },
5700 <pmn> \texttrademark = { 50,100},
5701 <ugm> \texttrademark = {150, 50},
5702 <ugm> \textcent = { 50, },
5703 <ugm> \textsterling = { , 50},
5704 <ugm> \textbrokenbar = {200,300},
5705 <pmn> \textasciidieresis = {300,200},
5706 <cmr> \textcopyright = {100, },
5707 <pad> \textcopyright = {200,100},
5708 <pmn> \textcopyright = {100,150},
5709 <ugm> \textcopyright = {300, },
5710 <cmr> \textordfeminine = {100,100},
5711 <pmn> \textordfeminine = {200,200},
5712 <ugm> \textordfeminine = {100,200},
5713 <cmr|pad> \textlnot = {300, },
5714 <pmn|ugm> \textlnot = {200, },
5715 <cmr> \textregistered = {100, },
5716 <pad> \textregistered = {200,100},
5717 <pmn> \textregistered = { 50,150},
5718 <ugm> \textregistered = {300, },
5719 <pmn> \textasciimacron = {150,200},
5720 <cmr|pad> \textdegree = {500,100},
5721 <pmn> \textdegree = {150,150},
5722 <ugm> \textdegree = {300,200},
5723 <cmr> \textpm = {150,100},
5724 <pad> \textpm = {200,150},
5725 <pmn|ugm> \textpm = {150,200},
5726 <cmr> \textonesuperior = {400, },
5727 <pad> \textonesuperior = {300,100},
5728 <pmn> \textonesuperior = {200,100},
5729 <ugm> \textonesuperior = {300,300},
5730 <cmr> \texttwosuperior = {400, },
5731 <pad> \texttwosuperior = {300, },
5732 <pmn> \texttwosuperior = {200,100},
5733 <ugm> \texttwosuperior = {300,200},
5734 <cmr> \textthreesuperior = {400, },

```

```

5735 <pad> \textthreesuperior = {300, },
5736 <pmn> \textthreesuperior = {200,100},
5737 <ugm> \textthreesuperior = {300,200},
5738 <ugm> \textmu = { ,100},
5739 <pmn> \textasciicute = {300,200},
5740 <cmr> \textparagraph = {200, },
5741 <pmn> \textparagraph = { ,100},
5742 <cmr> \textperiodcentered = {500,500},
5743 <pad|pmn|ugm> \textperiodcentered = {300,400},
5744 <cmr> \textordmasculine = {100,100},
5745 <pmn> \textordmasculine = {200,200},
5746 <ugm> \textordmasculine = {300,200},
5747 <cmr> \texteuro = {200, },
5748 <pad> \texteuro = {100, },
5749 <pmn> \texteuro = {100,-50},
5750 <cmr> \texttimes = {200,200},
5751 <pad> \texttimes = {200,100},
5752 <pmn> \texttimes = { ,70,100},
5753 <ugm> \texttimes = {200,300},
5754 <cmr|pad> \textdiv = {200,200}
5755 <pmn> \textdiv = {150,200}
5756 <ugm> \textdiv = {200,300},
5757 <ugm> \textsection = { ,200},
5758 <ugm> \textonehalf = { 50,100},
5759 <ugm> \textonequarter = { 50,100},
5760 <ugm> \textthreequarters = { 50,100},
5761 <ugm> \textsurd = { ,100}
5762 }
5763
5764 </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}

```

```

5765 <*cmr>
5766 \SetProtrusion
5767 [ name = cmr-math-letters ]
5768 { encoding = OML,
5769   family = cmm,
5770   series = {m,b},
5771   shape = it }
5772 {
5773   A = {100, 50}, % \mathnormal
5774   B = { 50, },
5775   C = { 50, },

```

```

5776     D = { 50, 50},
5777     E = { 50,   },
5778     F = {100, 50},
5779     G = { 50, 50},
5780     H = { 50, 50},
5781     I = { 50, 50},
5782     J = {150, 50},
5783     K = { 50,100},
5784     L = { 50, 50},
5785     M = { 50,   },
5786     N = { 50,   },
5787     O = { 50,   },
5788     P = { 50,   },
5789     Q = { 50, 50},
5790     R = { 50,   },
5791     S = { 50,   },
5792     T = { 50,100},
5793     U = { 50, 50},
5794     V = {100,100},
5795     W = { 50,100},
5796     X = { 50,100},
5797     Y = {100,100},
5798     f = {100,100},
5799     h = {   ,100},
5800     i = {   , 50},
5801     j = {   , 50},
5802     k = {   , 50},
5803     r = {   , 50},
5804     v = {   , 50},
5805     w = {   , 50},
5806     x = {   , 50},
5807     "0B = { 50,100}, % \alpha
5808     "0C = { 50, 50}, % \beta
5809     "0D = {200,150}, % \gamma
5810     "0E = { 50, 50}, % \delta
5811     "0F = { 50, 50}, % \epsilon
5812     "10 = { 50,150}, % \zeta
5813     "12 = { 50,   }, % \theta
5814     "13 = {   ,100}, % \iota
5815     "14 = {   ,100}, % \kappa
5816     "15 = {100, 50}, % \lambda
5817     "16 = {   , 50}, % \mu
5818     "17 = {   , 50}, % \nu
5819     "18 = {   , 50}, % \xi
5820     "19 = { 50,100}, % \pi
5821     "1A = { 50, 50}, % \rho
5822     "1B = {   ,150}, % \sigma
5823     "1C = { 50,150}, % \tau
5824     "1D = { 50, 50}, % \upsilon
5825     "1F = { 50,100}, % \chi
5826     "20 = { 50, 50}, % \psi
5827     "21 = {   , 50}, % \omega
5828     "22 = {   , 50}, % \varepsilon
5829     "23 = {   , 50}, % \vartheta
5830     "24 = {   , 50}, % \varpi
5831     "25 = {100,   }, % \varrho
5832     "26 = {100,100}, % \varsigma
5833     "27 = { 50, 50}, % \varphi
5834     "28 = {100,100}, % \leftharpoonup
5835     "29 = {100,100}, % \leftharpoondown
5836     "2A = {100,100}, % \rightharpoonup
5837     "2B = {100,100}, % \rightharpoondown
5838     "2C = {300,200}, % \hook

```



```

5839 "2D = {200,300}, % \rhook
5840 "2E = { ,100}, % \triangleright
5841 "2F = {100, }, % \triangleleft
5842 "3A = { ,500}, % ., \ldotp
5843 "3B = { ,500}, % ,
5844 "3C = {200,100}, % <
5845 "3D = {300,400}, % /
5846 "3E = {100,200}, % >
5847 "3F = {200,200}, % \star
5848 "5B = { ,100}, % \flat
5849 "5E = {200,200}, % \smile
5850 "5F = {200,200}, % \frown
5851 "7C = {100, }, % \jmath
5852 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

5853 }
5854

```

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}

```

```

5855 \SetProtrusion
5856 [ name = cmr-math-symbols ]
5857 { encoding = OMS,
5858   family = cmsy,
5859   series = {m,b},
5860   shape = n }
5861 {
5862   A = {150, 50}, % \mathcal
5863   C = { ,100},
5864   D = { , 50},
5865   F = { 50,150},
5866   I = { ,100},
5867   J = {100,150},
5868   K = { ,100},
5869   L = {100, },
5870   M = { 50, 50},
5871   N = { 50,100},
5872   P = { , 50},
5873   Q = { 50, },
5874   R = { , 50},
5875   T = { 50,150},
5876   V = { 50, 50},
5877   W = { , 50},
5878   X = {100,100},
5879   Y = {100, },
5880   Z = {100,150},
5881   "00 = {300,300}, % -
5882   "01 = { ,700}, % \cdot, \cdotp
5883   "02 = {150,250}, % \times
5884   "03 = {150,250}, % *, \ast
5885   "04 = {200,300}, % \div
5886   "05 = {150,250}, % \diamond
5887   "06 = {200,200}, % \pm
5888   "07 = {200,200}, % \mp
5889   "08 = {100,100}, % \oplus
5890   "09 = {100,100}, % \ominus
5891   "0A = {100,100}, % \otimes
5892   "0B = {100,100}, % \oslash
5893   "0C = {100,100}, % \odot

```

```

5894 "0D = {100,100}, % \bigcirc
5895 "0E = {100,100}, % \circ
5896 "0F = {100,100}, % \bullet
5897 "10 = {100,100}, % \asymp
5898 "11 = {100,100}, % \equiv
5899 "12 = {200,100}, % \subseteq
5900 "13 = {100,200}, % \supseteq
5901 "14 = {200,100}, % \leq
5902 "15 = {100,200}, % \geq
5903 "16 = {200,100}, % \preceq
5904 "17 = {100,200}, % \succeq
5905 "18 = {200,200}, % \sim
5906 "19 = {150,150}, % \approx
5907 "1A = {200,100}, % \subset
5908 "1B = {100,200}, % \supset
5909 "1C = {200,100}, % \ll
5910 "1D = {100,200}, % \gg
5911 "1E = {300,100}, % \prec
5912 "1F = {100,300}, % \succ
5913 "20 = {100,200}, % \leftarrow
5914 "21 = {200,100}, % \rightarrow
5915 "22 = {100,100}, % \uparrow
5916 "23 = {100,100}, % \downarrow
5917 "24 = {100,100}, % \leftrightarrows
5918 "25 = {100,100}, % \nearrow
5919 "26 = {100,100}, % \searrow
5920 "27 = {100,100}, % \simeq
5921 "28 = {100,100}, % \Leftarrow
5922 "29 = {100,100}, % \Rightarrow
5923 "2A = {100,100}, % \Uparrow
5924 "2B = {100,100}, % \Downarrow
5925 "2C = {100,100}, % \Leftrightarrow
5926 "2D = {100,100}, % \nrightarrow
5927 "2E = {100,100}, % \swarrow
5928 "2F = { ,100}, % \propto
5929 "30 = { ,400}, % \prime
5930 "31 = {100,100}, % \infty
5931 "32 = {150,100}, % \in
5932 "33 = {100,150}, % \ni
5933 "34 = {100,100}, % \triangle, \bigtriangleup
5934 "35 = {100,100}, % \bigtriangledown
5935 "38 = { ,100}, % \forall
5936 "39 = {100, }, % \exists
5937 "3A = {200, }, % \neg
5938 "3E = {200,200}, % \top
5939 "3F = {200,200}, % \bot, \perp
5940 "5E = {100,200}, % \wedge
5941 "5F = {100,200}, % \vee
5942 "60 = { ,300}, % \vdash
5943 "61 = {300, }, % \dashv
5944 "62 = {100,100}, % \lfloor
5945 "63 = {100,100}, % \rfloor
5946 "64 = {100,100}, % \lceil
5947 "65 = {100,100}, % \rceil
5948 "66 = {150, }, % \lbrace
5949 "67 = { ,150}, % \rbrace
5950 "68 = {400, }, % \langle
5951 "69 = { ,400}, % \rangle
5952 "6C = {100,100}, % \updownarrow
5953 "6D = {100,100}, % \Updownarrow
5954 "6E = {100,300}, % \, \backslash, \setminus
5955 "72 = {100,100}, % \nabla
5956 "79 = {200,200}, % \dagger

```

```

5957 "7A = {100,100}, % \ddagger
5958 "7B = {100, }, % \mathparagraph
5959 "7C = {100,100}, % \clubsuit
5960 "7D = {100,100}, % \diamondsuit
5961 "7E = {100,100}, % \heartsuit
5962 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

5963 }
5964

```

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}
```

```

5965 </cmr>
5966 </cfg-t>

```

## 15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
5967 <*cfg-u>
```

Symbol font 'a'.

```

5968 <*msa>
5969 \SetProtrusion
5970 [ name = AMS-a ]
5971 { encoding = U,
5972   family = msa }
5973 {
5974   "05 = {150,250}, % \centerdot
5975   "06 = {100,100}, % \lozenge
5976   "07 = { 50, 50}, % \blacklozenge
5977   "08 = { 50, 50}, % \circlearrowright
5978   "09 = { 50, 50}, % \circlearrowleft
5979   "0A = {100,100}, % \rightleftharpoons
5980   "0B = {100,100}, % \leftrightharpoons
5981   "0D = {-50,200}, % \Vdash
5982   "0E = {-50,200}, % \Vvdash
5983   "0F = {-70,150}, % \vDash
5984   "10 = {100,150}, % \twoheadrightarrow
5985   "11 = {100,150}, % \twoheadleftarrow
5986   "12 = { 50,100}, % \leftleftarrows
5987   "13 = { 50, 80}, % \rightrightarrows
5988   "14 = {120,120}, % \upuparrows
5989   "15 = {120,120}, % \downdownarrows
5990   "16 = {200,200}, % \upharpoonright
5991   "17 = {200,200}, % \downharpoonright
5992   "18 = {200,200}, % \upharpoonleft
5993   "19 = {200,200}, % \downharpoonleft
5994   "1A = { 80,100}, % \rightarrowtail
5995   "1B = { 80,100}, % \leftarrowtail
5996   "1C = { 50, 50}, % \leftrightarrows
5997   "1D = { 50, 50}, % \rightleftarrows
5998   "1E = {250, }, % \Lsh
5999   "1F = { ,250}, % \Rsh
6000   "20 = {100,100}, % \rightsquigarrow
6001   "21 = {100,100}, % \leftrightsquigarrow
6002   "22 = {100, 50}, % \looparrowleft
6003   "23 = { 50,100}, % \looparrowright

```

```

6004 "24 = { 50, 80}, % \circeq
6005 "25 = { ,100}, % \succsim
6006 "26 = { ,100}, % \gtrsim
6007 "27 = { ,100}, % \gtrapprox
6008 "28 = {150, 50}, % \multimap
6009 "2B = {100,150}, % \doteqdot
6010 "2C = {100,150}, % \triangleq
6011 "2D = {100, 50}, % \precsim
6012 "2E = {100, 50}, % \lesssim
6013 "2F = { 50, 50}, % \lessapprox
6014 "30 = {100, 50}, % \eqslantless
6015 "31 = { 50, 50}, % \eqslantgtr
6016 "32 = {100, 50}, % \curlyeqprec
6017 "33 = { 50,100}, % \curlyeqsucc
6018 "34 = {100, 50}, % \preccurlyeq
6019 "36 = { 50, }, % \leqslant
6020 "38 = { , 50}, % \backprime
6021 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
6022 "3C = { 50,100}, % \succcurlyeq
6023 "3E = { , 50}, % \geqslant
6024 "40 = { , 50}, % \sqsubset
6025 "41 = { 50, }, % \sqsupset
6026 "42 = { ,150}, % \vartriangleright, \rhd
6027 "43 = {150, }, % \vartriangleleft, \lhd
6028 "44 = { ,100}, % \trianglerighteq, \unrhd
6029 "45 = {100, }, % \trianglelefteq, \unlhd
6030 "46 = {100,100}, % \bigstar
6031 "48 = { 50, 50}, % \blacktriangledown
6032 "49 = { ,100}, % \blacktriangleright
6033 "4A = {100, }, % \blacktriangleleft
6034 "4B = { ,150}, % \dashrightarrow (the arrow)
6035 "4C = {150, }, % \dashleftarrow
6036 "4D = { 50, 50}, % \vartriangle
6037 "4E = { 50, 50}, % \blacktriangle
6038 "4F = { 50, 50}, % \triangledown
6039 "50 = { 50, 50}, % \eqcirc
6040 "56 = { ,150}, % \Rrightarrow
6041 "57 = {150, }, % \Lleftarrow
6042 "58 = {100,300}, % \checkmark
6043 "5C = { 50, 50}, % \angle
6044 "5D = { 50, 50}, % \measuredangle
6045 "5E = { 50, 50}, % \sphericalangle
6046 "5F = { , 50}, % \varpropto
6047 "60 = {100,100}, % \smallsmile
6048 "61 = {100,100}, % \smallfrown
6049 "62 = { 50, }, % \Subset
6050 "63 = { , 50}, % \Supset
6051 "66 = {150,150}, % \curlywedge
6052 "67 = {150,150}, % \curlyvee
6053 "68 = { 50,150}, % \leftthreetimes
6054 "69 = {100, 50}, % \rightthreetimes
6055 "6C = { 50, 50}, % \bumpeq
6056 "6D = { 50, 50}, % \Bumpeq
6057 "6E = {100, }, % \lll
6058 "6F = { ,100}, % \ggg
6059 "70 = { 50,100}, % \ulcorner
6060 "71 = {100, 50}, % \urcorner
6061 "75 = {150,200}, % \dotplus
6062 "76 = { 50,100}, % \backsimeq
6063 "78 = { 50,100}, % \llcorner
6064 "79 = {100, 50}, % \lrcorner
6065 "7C = {100,100}, % \intercal
6066 "7D = { 50, 50}, % \circledcirc

```

```

6067 "7E = { 50, 50}, % \circledast
6068 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

6069 }
6070
6071 </msa>

```

Symbol font ‘b’.

```

6072 <*msb>
6073 \SetProtrusion
6074 [ name = AMS-b ]
6075 { encoding = U,
6076   family = msb }
6077 {
6078   A = { 50, 50}, % \mathbb
6079   C = { 50, 50},
6080   G = { , 50},
6081   L = { , 50},
6082   P = { , 50},
6083   R = { , 50},
6084   T = { , 50},
6085   V = { 50, 50},
6086   X = { 50, 50},
6087   Y = { 50, 50},
6088   "00 = { 50, 50}, % \lvertneqq
6089   "01 = { 50, 50}, % \gvertneqq
6090   "02 = { 50, 50}, % \nleq
6091   "03 = { 50, 50}, % \ngeq
6092   "04 = {100, 50}, % \less
6093   "05 = { 50,150}, % \ngtr
6094   "06 = {100, 50}, % \nprec
6095   "07 = { 50,150}, % \nsucc
6096   "08 = { 50, 50}, % \lneqq
6097   "09 = { 50, 50}, % \gneqq
6098   "0A = {100,100}, % \nleqslant
6099   "0B = {100,100}, % \ngeqslant
6100   "0C = {100, 50}, % \lneq
6101   "0D = { 50,100}, % \gneq
6102   "0E = {100, 50}, % \npreceq
6103   "0F = { 50,100}, % \nsucceq
6104   "10 = { 50, }, % \precnsim
6105   "11 = { 50, 50}, % \succnsim
6106   "12 = { 50, 50}, % \lnsim
6107   "13 = { 50, 50}, % \gnsim
6108   "14 = { 50, 50}, % \lneqq
6109   "15 = { 50, 50}, % \ngeqq
6110   "16 = { 50, 50}, % \precneqq
6111   "17 = { 50, 50}, % \succneqq
6112   "18 = { 50, 50}, % \precnapprox
6113   "19 = { 50, 50}, % \succnapprox
6114   "1A = { 50, 50}, % \lnapprox
6115   "1B = { 50, 50}, % \gnapprox
6116   "1C = {150,200}, % \nsim
6117   "1D = { 50, 50}, % \ncong
6118   "1E = {100,150}, % \diagup
6119   "1F = {100,150}, % \diagdown
6120   "20 = {100, 50}, % \varsubsetneq
6121   "21 = { 50,100}, % \varsupsetneq
6122   "22 = {100, 50}, % \subsetneqq
6123   "23 = { 50,100}, % \supsetneqq
6124   "24 = {100, 50}, % \subsetneqq
6125   "25 = { 50,100}, % \supsetneqq

```

```

6126 "26 = {100, 50}, % \varsubsetneqq
6127 "27 = { 50,100}, % \varsupsetneqq
6128 "28 = {100, 50}, % \subsetneq
6129 "29 = { 50,100}, % \supsetneq
6130 "2A = {100, 50}, % \nsubseteq
6131 "2B = { 50,100}, % \nsupseteq
6132 "2C = { 50,100}, % \nparallel
6133 "2D = {100,150}, % \nmid
6134 "2E = {150,150}, % \nshortmid
6135 "2F = {100,100}, % \nshortparallel
6136 "30 = { ,150}, % \nvdash
6137 "31 = { ,150}, % \nVdash
6138 "32 = { ,100}, % \nvDash
6139 "33 = { ,100}, % \nVDash
6140 "34 = { ,100}, % \ntrianglerighteq
6141 "35 = {100, }, % \trianglelefteq
6142 "36 = {100, }, % \triangleleft
6143 "37 = { ,100}, % \triangleright
6144 "38 = {100,200}, % \leftarrow
6145 "39 = {100,200}, % \rightarrow
6146 "3A = {100,100}, % \Leftarrow
6147 "3B = { 50,100}, % \Rightarrow
6148 "3C = {100,100}, % \Leftrightarrow
6149 "3D = {100,200}, % \leftrightarrows
6150 "3E = { 50, 50}, % \divideontimes
6151 "3F = { 50, 50}, % \varnothing
6152 "60 = {200, }, % \Finv
6153 "61 = { , 50}, % \Game
6154 "68 = {100,100}, % \eqsim
6155 "69 = { 50, }, % \beth
6156 "6A = { 50, }, % \gimel
6157 "6B = {150, }, % \daleth
6158 "6C = {200, }, % \lessdot
6159 "6D = { ,200}, % \gtrdot
6160 "6E = {100,200}, % \ltimes
6161 "6F = {150,100}, % \rtimes
6162 "70 = { 50,100}, % \shortmid
6163 "71 = { 50, 50}, % \shortparallel
6164 "72 = {200,300}, % \smallsetminus
6165 "73 = {100,200}, % \thicksim
6166 "74 = { 50,100}, % \thickapprox
6167 "75 = { 50, 50}, % \approx
6168 "76 = { 50,100}, % \succapprox
6169 "77 = { 50, 50}, % \precapprox
6170 "78 = {100,100}, % \curvearrowleft
6171 "79 = { 50,150}, % \curvearrowright
6172 "7A = { 50,200}, % \digamma
6173 "7B = {100, 50}, % \varkappa
6174 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

6175 }
6176
6177 </msb>

```

## 15.8.8 Euler

Euler Roman font (package `euler`).

```

6178 <*eur>
6179 \SetProtrusion
6180 [ name = euler ]

```

```

6181 { encoding = U,
6182     family  = eur }
6183 {
6184     "01 = {100,100},
6185     "03 = {100,150},
6186     "06 = {    ,100},
6187     "07 = {100,150},
6188     "08 = {100,100},
6189     "0A = {100,100},
6190     "0B = {    , 50},
6191     "0C = {    ,100},
6192     "0D = {100,100},
6193     "0E = {    ,100},
6194     "0F = {100,100},
6195     "10 = {100,100},
6196     "13 = {    ,100},
6197     "14 = {    ,100},
6198     "15 = {    , 50},
6199     "16 = {    , 50},
6200     "17 = { 50,100},
6201     "18 = { 50,100},
6202     "1A = {    , 50},
6203     "1B = {    , 50},
6204     "1C = { 50,100},
6205     "1D = { 50,100},
6206     "1E = { 50,100},
6207     "1F = { 50,100},
6208     "20 = {    , 50},
6209     "21 = {    , 50},
6210     "22 = { 50,100},
6211     "24 = {    , 50},
6212     "27 = { 50,100},
6213     1 = {100,100},
6214     7 = { 50,100},
6215     "3A = {300,500},
6216     "3B = {200,400},
6217     "3C = {200,100},
6218     "3D = {200,200},
6219     "3E = {100,200},
6220     A = {    ,100},
6221     D = {    , 50},
6222     J = { 50,    },
6223     K = {    , 50},
6224     L = {    , 50},
6225     Q = {    , 50},
6226     T = { 50,    },
6227     X = { 50, 50},
6228     Y = { 50,    },
6229     h = {    , 50},
6230     k = {    , 50}
6231 }
6232

```

Extended by the `eulervm` package.

```

6233 \SetProtrusion
6234 [ name      = euler-vm,
6235   load      = euler ]
6236 { encoding = U,
6237   family   = zeur }
6238 {
6239     "28 = {100,200},
6240     "29 = {100,200},
6241     "2A = {100,150},

```

```

6242     "2B = {100,150},
6243     "2C = {200,300},
6244     "2D = {200,300},
6245     "2E = {    ,100},
6246     "2F = {100,   },
6247     "3F = {150,150},
6248     "5B = {    ,100},
6249     "5E = {100,100},
6250     "5F = {100,100},
6251     "80 = {    , 50},
6252     "81 = {200,250},
6253     "82 = {100,200}
6254 }
6255
6256 (/eur)

```

Euler Script font (euca1).

```

6257 (*eus)
6258 \SetProtrusion
6259 [ name      = euscript ]
6260 { encoding = U,
6261   family   = eus   }
6262 {
6263     A = {100,100},
6264     B = { 50,100},
6265     C = { 50, 50},
6266     D = { 50,100},
6267     E = { 50,100},
6268     F = { 50,   },
6269     G = { 50,   },
6270     H = {    ,100},
6271     K = {    , 50},
6272     L = {    ,150},
6273     M = {    , 50},
6274     N = {    , 50},
6275     O = { 50, 50},
6276     P = { 50, 50},
6277     T = {    ,100},
6278     U = {    , 50},
6279     V = { 50, 50},
6280     W = { 50, 50},
6281     X = { 50, 50},
6282     Y = { 50,   },
6283     Z = { 50,100},
6284     "00 = {250,250},
6285     "18 = {200,200},
6286     "3A = {200,150},
6287     "40 = {    ,100},
6288     "5E = {100,100},
6289     "5F = {100,100},
6290     "66 = { 50,   },
6291     "67 = {    , 50},
6292     "6E = {200,200}
6293 }
6294
6295 \SetProtrusion
6296 [ name      = euscript-vm,
6297   load      = euscript ]
6298 { encoding = U,
6299   family   = zeus   }
6300 {
6301     "01 = {600,600},
6302     "02 = {200,200},

```



```
6303 "03 = {200,200},
6304 "04 = {200,200},
6305 "05 = {150,150},
6306 "06 = {200,200},
6307 "07 = {200,200},
6308 "08 = {100,100},
6309 "09 = {100,100},
6310 "0A = {100,100},
6311 "0B = {100,100},
6312 "0C = {100,100},
6313 "0D = {100,100},
6314 "0E = {150,150},
6315 "0F = {100,100},
6316 "10 = {150,150},
6317 "11 = {100,100},
6318 "12 = {150,100},
6319 "13 = {100,150},
6320 "14 = {150,100},
6321 "15 = {100,150},
6322 "16 = {200,100},
6323 "17 = {100,200},
6324 "19 = {150,150},
6325 "1A = {150,100},
6326 "1B = {100,150},
6327 "1C = {100,100},
6328 "1D = {100,100},
6329 "1E = {250,100},
6330 "1F = {100,250},
6331 "20 = {150,200},
6332 "21 = {150,200},
6333 "22 = {150,150},
6334 "23 = {150,150},
6335 "24 = {100,200},
6336 "25 = {150,150},
6337 "26 = {150,150},
6338 "27 = {100,100},
6339 "28 = {100,100},
6340 "29 = {100,150},
6341 "2A = {100,100},
6342 "2B = {100,100},
6343 "2C = {100,100},
6344 "2D = {150,150},
6345 "2E = {150,150},
6346 "2F = {100,100},
6347 "30 = {100,100},
6348 "31 = {100,100},
6349 "32 = {100,100},
6350 "33 = {100,100},
6351 "34 = {100,100},
6352 "35 = {100,100},
6353 "3E = {150,150},
6354 "3F = {150,150},
6355 "60 = { ,200},
6356 "61 = {200, },
6357 "62 = {100,100},
6358 "63 = {100,100},
6359 "64 = {100,100},
6360 "65 = {100,100},
6361 "68 = {300, },
6362 "69 = { ,300},
6363 "6C = {100,100},
6364 "6D = {100,100},
6365 "6F = {100,100},
```

```

6366 "72 = {100,100},
6367 "73 = {200,100},
6368 "76 = { ,100},
6369 "77 = {100, },
6370 "78 = { 50, 50},
6371 "79 = {100,100},
6372 "7A = {100,100},
6373 "7D = {150,150},
6374 "7E = {100,100},
6375 "A8 = {100,100},
6376 "A9 = {100,100},
6377 "AB = {200,200},
6378 "BA = { ,200},
6379 "BB = { ,200},
6380 "BD = {200,200},
6381 "DE = {200,200}
6382 }
6383
6384 (/eus)

```

Euler Fraktur font (eufrak).

```

6385 (*euf)
6386 \SetProtrusion
6387 [ name = mathfrak ]
6388 { encoding = U,
6389   family = euf }
6390 {
6391   A = { , 50},
6392   B = { , 50},
6393   C = { 50, 50},
6394   D = { , 80},
6395   E = { 50, },
6396   G = { , 50},
6397   L = { , 80},
6398   O = { , 50},
6399   T = { , 80},
6400   X = { 80, 50},
6401   Z = { 80, 50},
6402   b = { , 50},
6403   c = { , 50},
6404   k = { , 50},
6405   p = { , 50},
6406   q = { 50, },
6407   v = { , 50},
6408   w = { , 50},
6409   x = { , 50},
6410   1 = {100,100},
6411   2 = { 80, 80},
6412   3 = { 80, 50},
6413   4 = { 80, 50},
6414   7 = { 50, 50},
6415   "12 = {500,500},
6416   "13 = {500,500},
6417   ! = { ,200},
6418   ' = {200,300},
6419   ( = {200, },
6420   ) = { ,200},
6421   * = {200,200},
6422   + = {200,250},
6423   - = {200,200},
6424   {,} = {300,300},
6425   . = {400,400},
6426   {=} = {200,200},

```

```

6427      : = { ,200},
6428      ; = { ,200},
6429      ] = { ,200}
6430    }
6431
6432 </euf>
6433 </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>).

```

6434 <*cfg-e>
6435 \SetProtrusion
6436 <zpeu|euroitc> { encoding = U,
6437 <mvs> { encoding = {OT1,U},
6438 <zpeu> family = zpeu }
6439 <euroitc> family = {euroitc,euroitcs} }
6440 <mvs> family = mvs }
6441 {
6442 <zpeu> E = {50, }
6443 <euroitc> E = {100,50}
6444 <mvs> 164 = {50,50}, % \EUR
6445 <mvs> 068 = {50,-100} % \EURdig
6446 }
6447
6448 <*zpeu|euroitc>
6449 \SetProtrusion
6450 { encoding = U,
6451 <zpeu> family = zpeu,
6452 <euroitc> family = {euroitc,euroitcs},
6453 shape = it* }
6454 {
6455 <zpeu> E = {100,-50}
6456 <euroitc> E = {100,}
6457 }
6458
6459 </zpeu|euroitc>
6460 <*zpeu>
6461 \SetProtrusion
6462 { encoding = U,
6463 family = {zpeus,eurosans} }
6464 {
6465 E = {100,50}
6466 }
6467
6468 \SetProtrusion
6469 { encoding = U,
6470 family = {zpeus,eurosans},
6471 shape = it* }
6472 {
6473 E = {200, }
6474 }
6475
6476 </zpeu>
6477 </cfg-e>

```

<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.

2      6      7      5      3      4      1

Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei

## 15.9 Interword Spacing

Default unit is space.

```
6478 (*m-t)
6479 %%% -----
6480 %%% INTERWORD SPACING
6481
6482 \SetExtraSpacing
6483 [ name = default ]
6484 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6485 {
```

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

```
6486 { , } = { , -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)

```
6487 r = { , -300, 300 },
```

- [before or] after lowercase characters with ascenders

```
6488 b = { , -200, 200 },
6489 d = { , -200, 200 },
6490 f = { , -200, 200 },
6491 h = { , -200, 200 },
6492 k = { , -200, 200 },
6493 l = { , -200, 200 },
6494 t = { , -200, 200 },
```

- [before or] after lowercase characters with x-height plus descender with additional optical space, e. g., ‘v’, or ‘w’

```
6495 c = { , -100, 100 },
```

```

6496      p = { , -100, 100 },
6497      v = { , -100, 100 },
6498      w = { , -100, 100 },
6499      z = { , -100, 100 },
6500      x = { , -100, 100 },
6501      y = { , -100, 100 },

```

- [before or] after lowercase characters with x-height plus descender without additional optical space

```

6502      i = { , 50, -50 },
6503      m = { , 50, -50 },
6504      n = { , 50, -50 },
6505      u = { , 50, -50 },

```

- after colon and semicolon

```

6506      : = { , 200, -200 },
6507      ; = { , 200, -200 },

```

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

```

6508      . = { , 250, -250 },
6509      ! = { , 250, -250 },
6510      ? = { , 250, -250 }

```

The order has to be reversed when enlarging is needed.'

```

6511      }
6512

```

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 `TEXbook`:

'If the space factor  $f$  is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if  $f \geq 2000$ . [...] Then the stretch component is multiplied by  $f/1000$ , while the shrink component is multiplied by  $1000/f$ .'

The 'extra space' (`\fontdimen7`) for Computer Modern Roman is a third of `\fontdimen2`, i. e., 333.

```

6513 \SetExtraSpacing
6514 [ name = nonfrench-cmr,
6515   load = default,
6516   context = nonfrench ]
6517 { encoding = {OT1,T1,LY1,OT4,QX,T5},
6518   family = cmr }
6519 {

```

latex.ltx has:

```

\def\nonfrenchspacing{
  \sfcode`. 3000

6520   . = {333,2000,-667},

      \sfcode`? 3000

6521   ? = {333,2000,-667},

      \sfcode`! 3000

6522   ! = {333,2000,-667},

      \sfcode`: 2000

6523   : = {333,1000,-500},

      \sfcode`; 1500

6524   ; = {   , 500,-333},

      \sfcode`\, 1250

6525   {,}= {   , 250,-200}

      }

6526 }
6527
```

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.

```

6528 \SetExtraSpacing
6529 [ name    = nonfrench-default,
6530   load     = default,
6531   context  = nonfrench ]
6532 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6533 {
6534   . = {240,2000,-667},
6535   ? = {240,2000,-667},
6536   ! = {240,2000,-667},
6537   : = {240,1000,-500},
6538   ; = {   , 500,-333},
6539   {,}= {   , 250,-200}
6540 }
6541
```

## 15.10 Additional Kerning

Default unit is 1em.

```

6542 %%% -----
6543 %%% ADDITIONAL KERNING
6544
```

A dummy list to be loaded when no context is active.

```
6545 \SetExtraKerning
6546   [ name = empty ]
6547   { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
6548   { }
6549
```

### 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.)

```
6550 \SetExtraKerning
6551   [ name      = french-default,
6552     context   = french,
6553     unit      = space ]
6554   { encoding = {OT1,T1,LY1} }
6555   {
6556     : = {1000,}, % = \fontdimen2
6557     ; = {500, }, % ~ \thinspace
6558     ! = {500, },
6559     ? = {500, }
6560   }
6561
```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfT<sub>E</sub>X.

```
6562 \SetExtraKerning
6563   [ name      = french-guillemets,
6564     context   = french-guillemets,
6565     load      = french-default,
6566     unit      = space ]
6567   { encoding = {T1,LY1} }
6568   {
6569     \guillemotleft = { ,800}, % = 0.8\fontdimen2
6570     \guillemotright = {800, }
6571   }
6572
6573 \SetExtraKerning
6574   [ name      = french-guillemets-OT1,
6575     context   = french-guillemets,
6576     load      = french-default,
6577     unit      = space ]
6578   { encoding = OT1 }
6579   { }
6580
```

### 15.10.2 Turkish

```
6581 \SetExtraKerning
6582   [ name      = turkish,
```

<sup>16</sup> [http://fr.wikipedia.org/wiki/Espace\\_typographique](http://fr.wikipedia.org/wiki/Espace_typographique), 5 July 2007.

```

6583     context = turkish ]
6584   { encoding = {OT1,T1,LY1} }
6585   {
6586     : = {167, }, % = \thinspace
6587     ! = {167, },
6588     {=} = {167, }
6589   }
6590
6591 </m-t>
6592 </config>

```

## 16 Auxiliary File for Micro Fine Tuning

This file can be used to test protrusion and expansion settings.

```

6593 <*test>
6594 \documentclass{article}
6595
6596 %% Here you can specify the font you want to test, using
6597 %% the commands \fontfamily, \fontseries and \fontshape.
6598 %% Make sure to end all lines with a comment character!
6599 \newcommand*\TestFont{%
6600   \fontfamily{ppl}%
6601   %% \fontseries{b}%
6602   %% \fontshape{it}% sc, sl
6603 }
6604
6605 \usepackage{ifthen}
6606 \usepackage[T1]{fontenc}
6607 \usepackage[latin1]{inputenc}
6608 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
6609
6610 \pagestyle{empty}
6611 \setlength{\parindent}{0pt}
6612 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
6613 \newcommand*\testprotrusion[2][ ]{%
6614   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
6615   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6616   you know the rest%
6617   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6618   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6619   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6620   \linebreak
6621   {\fontencoding{\encodingdefault}%
6622    \fontseries{\seriesdefault}%
6623    \fontshape{\shapedefault}%
6624    \selectfont
6625    Here is the beginning of a line, \dotfill and here is its end}\linebreak
6626 }
6627 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
6628 \def\stripprefix#1>{}
6629 \newcount\charcount
6630 \begin{document}
6631
6632 \microtypesetup{expansion=false}
6633
6634 {\centering The font in this document is called by:\\
6635  \texttt{\showTestFont}\par}\bigskip
6636
6637 \TestFont\selectfont
6638 This line intentionally left empty\linebreak
6639 %% A -- Z
6640 \charcount=65

```



```

6641 \loop
6642   \testprotrusion{\char\charcount}
6643   \advance\charcount 1
6644   \ifnum\charcount < 91 \repeat
6645   %% a -- z
6646   \charcount=97
6647 \loop
6648   \testprotrusion{\char\charcount}
6649   \advance\charcount 1
6650   \ifnum\charcount < 123 \repeat
6651   %% 0 -- 9
6652   \charcount=48
6653 \loop
6654   \testprotrusion{\char\charcount}
6655   \advance\charcount 1
6656   \ifnum\charcount < 58 \repeat
6657   %%
6658   \testprotrusion[r]{,}
6659   \testprotrusion[r]{.}
6660   \testprotrusion[r]{;}
6661   \testprotrusion[r]{:}
6662   \testprotrusion[r]{?}
6663   \testprotrusion[r]{!}
6664   \testprotrusion[l]{\textexclamdown}
6665   \testprotrusion[l]{\textquestiondown}
6666   \testprotrusion[r]{}}
6667   \testprotrusion[l]{({}
6668   \testprotrusion{/}
6669   \testprotrusion{\char`\}
6670   \testprotrusion{-}
6671   \testprotrusion{\textendash}
6672   \testprotrusion{\textemdash}
6673   \testprotrusion{\textquotelleft}
6674   \testprotrusion{\textquoteright}
6675   \testprotrusion{\textquotedblleft}
6676   \testprotrusion{\textquotedblright}
6677   \testprotrusion{\quotesinglbase}
6678   \testprotrusion{\quotedblbase}
6679   \testprotrusion{\guilsinglleft}
6680   \testprotrusion{\guilsinglright}
6681   \testprotrusion{\guillemotleft}
6682   \testprotrusion{\guillemotright}
6683
6684 \newpage
6685 The following displays the current font stretched by 5%,
6686 normal, and shrunk by 5%:
6687
6688 \bigskip
6689 \newlength{\MTln}
6690 \newcommand*\teststring
6691   {ABCDEFGH IJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
6692 \settowidth{\MTln}{\teststring}
6693 \microtypesetup{expansion=true}
6694
6695 \parbox{1.05\MTln}{\teststring\linebreak\}
6696   \teststring\par\bigskip
6697 \parbox{0.95\MTln}{\teststring}
6698
6699 \end{document}
6700 /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’ . . . . .	79
issue an error instead of a warning, when pdfTeX version is too old for autoexpand . . . . .	120	<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> ) . . . . .	128	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i> ) . . . . .	42
Protrusion: add factors for some more characters . . . . .	133	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i> ) . . . . .	83
settings for Adobe Minion (contributed by <i>Harald Harders</i> ) . . . . .	134	<code>\MT@pdftex@no</code> : fix: version check (reported by <i>Harald Harders</i> ) . . . . .	37
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance . . . . .	104	<code>\MT@permute</code> : don’t use sets for empty encoding . . . . .	106
<code>\MT@declare@sets</code> : remove spaces around set name . . . . .	92	<code>\MT@split@codes</code> : fix: allow zero and negative values . . . . .	58
<code>\MT@DeclareSet</code> : remove spaces around first argument . . . . .	91	<code>\MT@use@set</code> : remove spaces around set name . . . . .	96
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded . . . . .	77	<code>\UseMicrotypeSet</code> : remove spaces around first argument . . . . .	96

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

Font Sets: declare cmor as an alias of cmr . . . . .	126	changed . . . . .	93
new: allmath and basicmath . . . . .	125	<code>\MT@get@inh@list</code> : fix: set inheritance list \globally to \empty . . . . .	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 \set@fontsize instead of \setfontsize . . . . .	94
<code>\MT@encoding@check</code> : check whether only one encoding specified . . . . .	105	<code>\MT@loop</code> : fix: new macro, used instead of \loop . . . . .	46
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement . . . . .	54	<code>\MT@maybe@do</code> : also check for alias font name . . . . .	55
<code>\MT@get@basefamily</code> : also remove ‘w’ (swash capitals) . . . . .	79	<code>\MT@permute@@@@@</code> : more sanity checks for \SetProtrusion and \SetExpansion . . . . .	107
<code>\MT@get@highlevel</code> : check whether defaults have		<code>\MT@setupfont</code> : also search for alias font file . . . . .	52
		fix: call \@@enc@update if necessary . . . . .	52

### Version 1.3 (2004/10/27)

General: fix: specifying load option does no longer require to give a name, too . . . . .	102	<code>\MT@fix@catcode</code> : check some category codes (compatibility with german) . . . . .	34
Font Sets: declare aer, zer and hfor as aliases of cmr . . . . .	126	<code>\MT@load@list</code> : check whether list exists . . . . .	77

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

General: check for pdfcprot . . . . .	50	(OT1, T1, lmr) . . . . .	139
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 \pickup@font instead of \define@newfont as the hook for \MT@setupfont . . . . .	87	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too . . . . .	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 .....	56
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	41	\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) .....	43
slanted like italics .....	142	\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> ) .....	113	\MT@cfg@catcodes: reset catcode of '=' (compatibility with Turkish babel) .....	78
defaults: calculate step as min(stretch,shrink)/5	119	\MT@fix@catcode: reset catcode of '~' (compatibility with chemsym) .....	34
defaults: turn off expansion for DVI output .....	118	\MT@get@highlevel: don't test defaults if called after begin document .....	93
disable automatic expansion for DVI output .....	120	\MT@scale@factor: warning for factors outside limits	60
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 .....	59
Documentation: add 'Short History' .....	29	\MT@set@ex@codes: allow non-selected font expansion .....	64
add note about DVIoutput option .....	9	\MT@set@pr@codes: adjust protrusion factors before setting the inheriting characters .....	57
Inheritance: remove \ss from T1 list, add \DJ .....	128		
Protrusion: settings for Bitstream Charter .....	134		
\DeclareMicrotypeAlias: remove spaces around arguments .....	97		

**Version 1.6 (2005/01/24)**

General: defaults: turn off expansion for old pdfTeX versions .....	114	improve settings for numbers (pointed out by <i>Peter Muthesius</i> ) .....	136
disable automatic expansion for old pdfTeX versions .....	120	tune CMR math letters (OML encoding) .....	159
load a font if none is selected .....	52	\MT@get@charwd: use e-TeX's \fontcharwd, if available	59
new option: factor, by default 1000 .....	113	\MT@get@inh@list: correct message if selected is false .....	81
restructure dtx file .....	125	\MT@set@ex@codes: introduce factor option .....	64
test whether \pickup@font has changed .....	89	\MT@set@pr@codes: introduce factor option .....	57
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 ..	42	\MT@vinfo: new macro: used instead of \ifMT@verbose .....	35
Protrusion: add italic uppercase Greek letters .....	142		

**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 .....	37
		\MT@reset@ef@codes: only reset \efcodes for older pdfTeX versions .....	64

**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 <code>\chardef</code> commands	82
modify <code>\showhyphens</code>	121	test whether <code>\(encoding)\(...</code> is defined	82
new value for verbose option: errors	113	<code>\MT@if@list@exists</code> : don't define <code>\MT@#1@ec@name</code>	
shorter command names	47	globally, here and elsewhere	80
warning when running in draft mode	118	<code>\MT@if@dimen</code> : comparison with 1 to allow size smaller	
Documentation: add hint about compatibility	26	than 1 (suggested by <i>Andreas Böhmann</i> )	43
remove table of match order	12	<code>\MT@increment</code> : use e-TeX's <code>\numexpr</code> if available	46
Protrusion: fix: remove <code>\</code> from OT1, add		<code>\MT@is@composite</code> : new macro: construct command	
<code>\textbackslash</code> to T1 encoding	137	for composite character; no uncontrolled expansion	85
<code>\DeclareMicrotypeAlias</code> : may also be used inside		<code>\MT@scale</code> : new macro: use e-TeX's <code>\numexpr</code> if avail-	
configuration files	97	able	47
<code>\LoadMicrotypeFile</code> : new command (suggested by		<code>\MT@set@ex@codes</code> : two versions of this macro	64
<i>Andreas Böhmann</i> )	97	<code>\MT@split@name</code> : don't define <code>\MT@encoding &amp;c.</code>	
<code>\Microtype@Hook</code> : new command for font package		globally	54
authors	115	<code>\MT@test@ast</code> : make it simpler	93
<code>\microtypesetup</code> : fix: warning also when setting to		<code>\MT@try@order</code> : always check for size, too (suggested	
(no)compatibility	115	by <i>Andreas Böhmann</i> )	79
<code>\MT@begin@catcodes</code> : also use inside configuration		fix: also check for <code>//(series)/(shape)//</code> (reported by	
commands	79	<i>Andreas Böhmann</i> )	79
<code>\MT@cfg@catcodes</code> : reset catcode of <code>'</code> (compatibility		<code>\MT@warn@code@too@large</code> : new macro: type out max-	
with french* packages)	78	imum protrusion factor	60
<code>\MT@get@listname@</code> : use <code>\etfor</code> ( <i>Andreas Böhmann's</i>		<code>\MT@warn@err</code> : new macro: for verbose=errors	35
idea)	79		
<code>\MT@get@slot</code> : remove backslash hack	81		

### Version 1.8 (2005/06/23)

General: <code>\SetProtrusion</code> : new key: unit	103	<code>\MT@find@file</code> : no longer wrap names in commands	77
if font substitution has occurred, set up the substi-		<code>\MT@get@charwd</code> : warning for missing (resp. zero-	
tute font, not the selected one	87	width) characters	59
new option: config to load a different main config-		<code>\MT@get@font@dimen@six</code> : new macro: test whether	
uration file	114	<code>\fontdimen 6</code> is defined	57
new option: unit, by default character	113	<code>\MT@get@listname@</code> : made recursive	79
Documentation: add example for factor option	13	<code>\MT@get@slot</code> : fix: expand active characters	82
add example of how to get rid of a widow (sugges-		test whether <code>\(encoding)\(...</code> is defined made more	
ted by <i>Adam Kucharczyk</i> )	15	robust	82
add hint about error messages	27	<code>\MT@get@unit</code> : new macro: get unit for codes	61
Font Sets: add U encoding to <code>allmath</code>	125	<code>\MT@in@list</code> : made recursive	46
declare <code>pxr</code> and <code>txr</code> as aliases of <code>ppl</code> resp. <code>ptm</code>	126	<code>\MT@is@active</code> : new macro: translate inputenc-	
Inheritance: remove <code>\DJ</code> from T1 list (it's the same as		defined characters	83
<code>\DH</code> )	128	<code>\MT@is@letter</code> : warning for non-ASCII characters	83
Protrusion: add LY1 characters for Times	141	<code>\MT@led@kern</code> : character protrusion with <code>ledmac</code>	49
settings for AMS math fonts	163	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\etfor</code>	45
verified settings for slanted Computer Modern Ro-		<code>\MT@map@tlist@n</code> : new macro: used instead of <code>\etfor</code>	45
man	148	<code>\MT@old@cmd</code> : renamed commands from	
<code>\add@accent</code> : fix: disable micro-typographic setup in-		<code>\..MicroType..</code> to <code>\..Microtype..</code>	35
side <code>\add@accent</code> (reported by <i>Stephan Hennig</i> )	88	<code>\MT@pdfTeX@no</code> : case 5: pdfTeX 1.30	37
<code>\DeclareMicrotypeAlias</code> : warning when overriding		<code>\MT@permute@@@@@</code> : add ranges to the beginning of	
an alias font	97	the lists	108
<code>\DeclareMicrotypeSetDefault</code> : new command: set		<code>\MT@scale</code> : fix: remove spaces in e-TeX variant (repor-	
default font set	96	ted by <i>Mark Rossi</i> )	47
<code>\MT@cfg@catcodes</code> : reset catcodes of the remaining		<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when	
ASCII characters	78	<code>hyperref</code> is loaded	50
<code>\MT@check@rlist</code> : made recursive	108	restore csquotes's active characters	50
<code>\MT@curr@list@name</code> : new macro: current list type		restore percent character if Spanish <code>babel</code> is loaded	49
and name	85	<code>\MT@split@codes</code> : get character width once only	58
<code>\MT@declare@sets</code> : warning when redefining a set	92	<code>\MT@use@set</code> : fix: remove braces in first line	96
<code>\MT@define@set@key@</code> : use comma lists instead of		<code>\MT@xadd</code> : simplified	44
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 .....	90
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> .....	56
fix: use true as the default value .....	110	<code>\MT@detokenize@c</code> : fix the $\mathcal{E}$ - $\TeX$ version .....	42
option unit: rename value relative to character .....	113	<code>\MT@exp@two@n</code> : new macros: less <code>\expandafters</code> ..	41
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 ..	61
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 .....	8	<code>\MT@is@letter</code> : using <code>\catcode</code> should be more efficient than inspecting the <code>\meaning</code> .....	83
Font Sets: add OT4 encoding to text sets .....	125	<code>\MT@maybe@do</code> : redone .....	55
add T5 encoding to text sets .....	125	<code>\MT@pdf@tex@no</code> : compatibility with $\TeX$ Live hack (reported by <i>Herbert Voß</i> ) .....	38
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 .....	45
Inheritance: add list for OT4 .....	129	<code>\MT@scale@factor</code> : generalised .....	60
add list for T5 (requested by <i>Hàn Thê Thành</i> ) ..	131	<code>\MT@toks</code> : use instead of <code>\toks@</code> .....	39
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR .....	137	<code>\SetProtrusion</code> : (et al.) new key: font .....	99

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

General: ‘(file name)/(line number)’ as default list name .....	102	diately (requested by <i>Georg Verweyen</i> ) .....	92
new option: <code>defersetup</code> , by default true .....	112	<code>\MT@get@highlevel</code> : no longer check whether defaults have changed .....	93
remove superfluous test whether <code>\pickup@font</code> has changed .....	89	<code>\MT@ifdefined@c@T</code> : new macros: true case only ..	42
Documentation: add explanation for error message in DVI mode .....	27	<code>\MT@ifint</code> : use <code>\pdfmatch</code> if available .....	42
add explanation for error message with non-Type 1 fonts .....	27	<code>\MT@ifstreq</code> : use <code>\pdfstrcmp</code> if available .....	44
Font Sets: declare <code>mbch</code> (mathdesign) as an alias of Charter .....	127	<code>\MT@in@clist</code> : fix .....	45
Protrusion: fix: remove ‘_’ from OT1 encoding ..	138	<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) .....	60
settings for T5 encoded Charter .....	133	<code>\MT@is@feature</code> : new macro: check for pdf $\TeX$ feature .....	47
<code>\microtypesetup</code> : inside the preamble, accepts all package options .....	115	<code>\MT@map@clist@n</code> : following $\LaTeX$ 3 .....	45
<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 Verweyen</i> ) .....	92	<code>\MT@rem@from@clist</code> : fix .....	45
<code>\MT@get@highlevel</code> : no longer check whether defaults have changed .....	93	<code>\MT@setup@</code> : defer setup until the end of the preamble ..	48
<code>\MT@ifdefined@c@T</code> : new macros: true case only ..	42		
<code>\MT@ifint</code> : use <code>\pdfmatch</code> if available .....	42		
<code>\MT@ifstreq</code> : use <code>\pdfstrcmp</code> if available .....	44		
<code>\MT@in@clist</code> : fix .....	45		
<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) .....	60		
<code>\MT@is@feature</code> : new macro: check for pdf $\TeX$ feature .....	47		
<code>\MT@map@clist@n</code> : following $\LaTeX$ 3 .....	45		
<code>\MT@permute@@@@@</code> : don’t define permutations for unused encodings .....	107		
<code>\MT@rem@from@clist</code> : fix .....	45		
<code>\MT@setup@</code> : defer setup until the end of the preamble ..	48		

**Version 1.9b (2006/01/20)**

General: compatibility with <code>listings</code> : sanitise more catcodes (reported by <i>Holger Uhr</i> ) .....	51	add samples of micro-typographic features .....	4
compatibility with the <code>extendedchar</code> option of the <code>listings</code> package .....	51	<code>\MT@features</code> : use throughout the package to adjust to beta-ness .....	47
Documentation: activate expansion in the distributed PDF .....	1	<code>\MT@ifdimen</code> : use <code>\pdfmatch</code> if available .....	43
		<code>\MT@warn@code@too@large</code> : fix calculation with present factor .....	60

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

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verweyen</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 . . .	125	<code>\MT@get@font@dimen</code> : warning for zero <code>fontdimen</code> .	60
add QX encoding to text sets . . . . .	125	<code>\MT@get@opt</code> : optimise: don't reset when preset option is set . . . . .	61
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i> ) . . . . .	130	set list name before presetting . . . . .	61
Protrusion: settings for QX encoding (contributed by <i>Maciej Eder</i> ) . . . . .	139	<code>\MT@is@active</code> : support for Unicode (inputenc/utf8)	84
settings for Euro symbols (Adobe, ITC, marvosym)	171	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>tex4ht</code> is loaded (reported by <i>Peter Dybala</i> ) . . . . .	50
tweak AMS settings . . . . .	163	<code>\SetProtrusion</code> : (et al.) optimise: unify keys for mandatory argument . . . . .	99
<code>\DeclareCharacterInheritance</code> : fix: empty context	104	(et al.) split keys of optional and mandatory argument . . . . .	99
<code>\MT@detokenize@n</code> : new macro: use <code>\detokenize</code> if available . . . . .	42		
<code>\MT@get@ex@opt</code> : fix: evaluate preset . . . . .	65		

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

General: fix: default value for <code>activate</code> : <code>true</code> . . .	111	settings for Euler Roman font . . . . .	166
Documentation: add hint about unknown encodings include LPPL . . . . .	25	<code>\DeclareCharacterInheritance</code> : new key 'inputenc' to set the input encoding . . . . .	104
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> ) . . . . .	192	<code>\MT@rem@from@clist</code> : model after <code>\@removeelement</code>	45
Inheritance: adapt to <code>marvosym</code> 's changed encoding	126	<code>\MT@setup@</code> : empty <code>\MT@setup@</code> after use (compatibility with the <code>combine</code> class) . . . . .	48
Protrusion: complete settings for Euler Fraktur and Script fonts . . . . .	131	<code>\pickup@font</code> : no tracing with <code>trace</code> package . . . .	88
fix: forgotten comma in <code>mt-mvs.cfg</code> ; adapt to <code>marvosym</code> 's changed encoding . . . . .	170	<code>\SetExpansion</code> : new key: <code>inputenc</code> . . . . .	99
	171	<code>\SetProtrusion</code> : (et al.) new key: <code>inputenc</code> . . . .	99

**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 . . . . .	41	<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 . . . . .	99
<code>\MT@glet</code> : new macro . . . . .	41		

**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	113	compatibility with <code>listings</code> : set catcode of backslash to zero (reported by <i>Steven Bath</i> ) . . . . .	51
(beta:3) compatibility with <code>soul</code> : register <code>\textls</code> and <code>\sstyle</code> . . . . .	51	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 letterspacing commands only . . . . .	1
(beta:8) option 'babel': switch off Turkish <code>babel</code> 's		option 'unit', <code>\SetProtrusion</code> : deprecate value 'relative' completely . . . . .	103
		Documentation: add hint about how to increase <code>font_max</code> and <code>font_mem_size</code> . . . . .	27



add hint about warning when tracking <i>and</i> expansion is applied to a font	27	totally redone, using the new <code>\letterspacefont</code>	72
add remark about ‘draft’ option disabling microtype (noted by <i>Michalis Miatidis</i> )	9	<code>\MT@declare@sets</code> : fix: empty size list when redefining set	92
qualify hint about web documents with regard to older pdfTeX versions	25	<code>\MT@is@symbol</code> : made even more robust	84
qualify hints about expansion error messages with regard to older pdfTeX versions	27	<code>\MT@load@inputenc</code> : sanitise catcodes before loading input encoding (problem with listings)	62
Font Sets: add footnotesize and scriptsize sets	125	<code>\MT@pdfTeX@no</code> : (beta:1) case 6: pdfTeX 1.40	37
add smallcaps set	125	<code>\MT@split@name</code> : adjust to possible letterspacing	54
<code>\DeclareMicrotypeBabelHook</code> : (beta:1) new command: interaction with babel	98	<code>\SetExtraKerning</code> : (beta:1) new command: additional kerning	100
<code>\lsstyle</code> : (beta:1) new command: letterspacing	72	<code>\SetExtraSpacing</code> : (beta:1) new command: adjustment of interword spacing	100
(beta:8) fix: font switches don’t pose a problem anymore	72	<code>\SetTracking</code> : new command: tracking	100
(beta:8) fix: letterspacing commands may be nested	72	<code>\textls</code> : (beta:1) new command: letterspacing	75
		starred version: remove spaces around text	75
		<code>\tracingmicrotypeinpdf</code> : new debug method: mark all fonts with PDF annotations	36

### Version 2.1 (2007/01/21)

General: compatibility with CJK: also check for its definition	87	<code>\slig</code> : new command: protect ligatures in letter-spaced text	72
compatibility with pinyin: disable microtype in <code>\py@macron</code> (reported by <i>Sven Naumann</i> )	51	<code>\MT@get@ls@basefont</code> : redone: use <code>\pdfmatch</code> to make it bullet-proof	72
fix: letterspace package forgot to load keyval	39	<code>\textls</code> : fix: use <code>\hmode@bgroup</code>	75

### Version 2.2 (2007/07/14)

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i> )	48	<code>\MT@ifdimen</code> : employ LuaTeX features if available	43
new option: copyfonts	112	<code>\MT@ifint</code> : employ LuaTeX features if available	42
simplify key declarations	102	<code>\MT@ifstreq</code> : employ LuaTeX features if available	44
warning if stretch or shrink aren’t multiples of step	121	fix: <del>ε</del> -TeX version shouldn’t use <code>\x</code> and <code>\y</code> (found by <i>Wiebke Petersen</i> )	44
Documentation: add hint about error message with pdfTeX 1.40	27	<code>\MT@is@composite</code> : more robust: expand exactly once	85
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i> )	26	<code>\MT@is@symbol</code> : expand once more (for frenchpro)	84
add overview	6	<code>\MT@lsfont</code> : use <code>\font@name</code> , not <code>\MT@font</code>	69
logo transparency and amusement	1	<code>\MT@maybe@etex</code> : use catcode trickery	40
Font Sets: declare FPL Neu as an alias of Palatino	126	<code>\MT@pdfTeX@no</code> : case 7: pdfTeX 1.40.4	37
declare chr (chmath) as an alias of Charter (reported by <i>Geoff Vallis</i> )	127	<code>\MT@preset@aux@space</code> : generalised	63
default set for tracking: smallcaps	126	<code>\MT@requires@luatex</code> : (basic) support for LuaTeX	40
Inheritance: remove ‘-’ → ‘127’	129	<code>\MT@set@all@pr</code> : (et al.) allow empty values	57
Protrusion: settings for Bitstream Letter Gothic	134	<code>\MT@set@inputenc@</code> : only load inputenc files if necessary	62
Spacing: add sample	172	<code>\MT@set@tr@codes</code> : disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	70
Tracking: add ligatures that are to be disabled	132	possibility to customise interword spacing	70
<code>\DeclareMicrotypeVariants</code> : new command	97	<code>\MT@setupfont</code> : don’t call <code>\@enc@update</code> anymore	52
<code>\DisableLigatures</code> : new optional argument: disable selected ligatures only	98	only add features that are available with the respective pdfTeX	52
<code>\slig</code> : always defined	72	<code>\MT@setupfont@hook</code> : restore percent character if Galician babel is loaded	49
<code>\MT@checklist@font</code> : fix: construct font name from characteristics	56	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letter-spaced fonts	58
<code>\MT@copy@font</code> : optionally work on copies of fonts	53	<code>\MT@tracking</code> : remember fonts that shouldn’t be letterspaced	69
<code>\MT@get@basefamily</code> : redone, working on font names and suffixes of arbitrary length	79	<code>\MT@tracking@</code> : fix: tracking couldn’t be re-enabled	69
<code>\MT@get@charwd</code> : subtract letterspacing amount from width	59	<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	122
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	72	<code>\MT@with@babel@and@T</code> : also inspect class options	48
		<code>\pickup@font</code> : letterspace: setup inside group	88

<code>\SetTracking</code> : new key ‘no ligatures’ to disable ligatures of letterspaced fonts . . . . .	100	<code>ris</code> ) . . . . .	100
new keys ‘spacing’ and ‘outer spacing’ to adjust interword spacing (suggested by <i>Steven E. Harris</i> ) . . . . .		third argument may be empty . . . . .	100
		<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code> . . . . .	90

### 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> ) . . . . .	45
enable protrusion when tracking is enabled . . . . .	121	<code>\MT@is@active</code> : support for extended Unicode (inputenc/utf8x resp. ucs) – experimental . . . . .	84
fix: really switch off Turkish shorthands . . . . .	124	<code>\MT@noligatures</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> ) . . . . .	113	<code>\MT@plain</code> : letterspace: support for eplain/miniltx . . . . .	37
turned some warnings into errors . . . . .	113	<code>\MT@set@curr@eos</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 . . . . .	100
<code>\slig</code> : redone: extract outer kerns from current letterspacing amount . . . . .	72	<code>\MT@tr@outer@l</code> : only change pre outer space if it contains glue . . . . .	74
<code>\microtypecontext</code> : made robust (reported by <i>Stephan Hennig</i> ) . . . . .	90		
<code>\MT@begin@catcodes</code> : fix: don’t disable <code>\KV@sp@def</code> . . . . .	79		
<code>\MT@define@set@key@font</code> : font: single asterisk means normal font . . . . .	94		

### Version 2.3a (2008/02/29)

General: error messages if pdfTeX is too old for extensions . . . . .	122	<code>\MT@fix@catcode</code> : fix catcodes earlier, and also for the letterspace package . . . . .	34
fix test for soul under plain TeX . . . . .	51	<code>\MT@getkey</code> : fix: key=val in class options list . . . . .	118
Documentation: add hint about babel having to be loaded first . . . . .	27	<code>\MT@maybe@etex</code> : removed . . . . .	40
add table of available and enabled features . . . . .	7	<code>\MT@set@codes</code> : generalised . . . . .	58
mention soulutf8 . . . . .	29	<code>\MT@setupfont@hook</code> : restore percent character if Mexican babel is loaded . . . . .	49
Protrusion: adjust LMR quotation marks again . . . . .	139		

## B Index

Links (in blue) refer to the page where the corresponding entry is described (bold face) resp. occurs. Plain numbers (in black) refer to the code line where the corresponding entry is defined (underlined) resp. used.

<b>Options:</b>		stretch . . . . .	8
DVIoutput . . . . .	9	tracking . . . . .	7
activate . . . . .	6	unit . . . . .	8
auto . . . . .	8	verbose . . . . .	9
babel . . . . .	9		
config . . . . .	9	<b>User Commands:</b>	
draft . . . . .	9	<code>\DeclareCharacterInheritance</code> . . . . .	20
expansion . . . . .	6	<code>\DeclareMicrotypeAlias</code> . . . . .	22
factor . . . . .	7	<code>\DeclareMicrotypeBabelHook</code> . . . . .	23
final . . . . .	9	<code>\DeclareMicrotypeSet</code> . . . . .	10
kerning . . . . .	7	<code>\DeclareMicrotypeSet*</code> . . . . .	10
letterspace . . . . .	9	<code>\DeclareMicrotypeSetDefault</code> . . . . .	11
protrusion . . . . .	6	<code>\DeclareMicrotypeVariants</code> . . . . .	21
selected . . . . .	8	<code>\DeclareMicrotypeVariants*</code> . . . . .	21
shrink . . . . .	8	<code>\DisableLigatures</code> . . . . .	25
spacing . . . . .	7	<code>\LoadMicrotypeFile</code> . . . . .	22
step . . . . .	8	<code>\slig</code> . . . . .	24
		<code>\lsstyle</code> . . . . .	23



- \microtypecontext ..... 22
  - \microtypesetup ..... 10
  - \SetExpansion ..... 14
  - \SetExtraKerning ..... 19
  - \SetExtraSpacing ..... 18
  - \SetProtrusion ..... 13
  - \SetTracking ..... 16
  - \textls ..... 23
  - \textls\* ..... 23
  - \textmicrotypecontext ..... 22
  - \UseMicrotypeSet ..... 11
- @@@enc@update ..... 790
- A**
- a0poster (package) ..... 94
  - activate (option) ..... 6, 111, 182
  - \add@accent ..... 2287
  - ae (package) ..... 21, 126
  - amssymb (package) ..... 163
  - article (package) ..... 22
  - auto (option) ..... 8, 111
- B**
- babel (option) ..... 9, 23, 27, 30, 111, 182
  - babel (package) ..... 3, 5, 17, 23, 24, 27, 49, 78, 98, 123, 124, 127, 179, 180, 182–184
- C**
- \cf@encoding ..... 790
  - chemsym (package) ..... 179
  - chmath (package) ..... 21, 127, 183
  - CJK (package) ..... 51, 87, 183
  - cm-super (package) ..... 8
  - color (package) ..... 9, 118
  - combine (package) ..... 48, 182
  - config (option) ..... 9, 21, 31, 114, 180
  - contour (package) ..... 118
  - copyfonts (option) ..... 53, 111, 112, 183
  - crop (package) ..... 118
  - csquotes (package) ..... 28, 49, 50, 137, 180
  - \curr@fontshape ..... 2243, 2245, 2285
- D**
- \DeclareCharacterInheritance ..... 20, 41, 2995
  - \DeclareMicrotypeAlias ..... 22, 43, 70, 2668
  - \DeclareMicrotypeBabelHook ..... 23, 45, 64, 2721
  - \DeclareMicrotypeSet ..... 10, 32, 57, 59, 71, 2396
  - \DeclareMicrotypeSet\* ..... 10, 2396
  - \DeclareMicrotypeSetDefault ..... 11, 34, 61, 2627
  - \DeclareMicrotypeVariants ..... 21, 42, 58, 63, 2655
  - \DeclareMicrotypeVariants\* ..... 21
  - deferssetup (option) ..... 49, 111, 112, 115, 181
  - \define@newfont ..... 2242, 2244, 2247
  - \DisableLigatures ..... 25, 40, 62, 2701
  - \do@subst@correction ..... 2284
  - docstrip (package) ..... 33
  - draft (option) ..... 9, 111, 112, 183
  - dsfont (package) ..... 57
  - DVIoutput (option) ..... 9, 111, 179
- E**
- eco (package) ..... 21, 126
- F**
- \efcode ..... 1281, 1305, 1306, 1348, 1350
  - eplain (package) ..... 1, 24, 29, 33, 37, 48, 118, 184
  - eucal (package) ..... 168
  - eufrak (package) ..... 170
  - euler (package) ..... 85, 166, 182
  - eulervm (package) ..... 21, 126, 167, 182
  - euroitc (package) ..... 171
  - europs (package) ..... 171
  - eurosans (package) ..... 171
  - expansion (option) ..... 6, 7, 11, 110, 111, 118
- F**
- \f@encoding ..... 790
  - \f@size ..... 2243, 2245, 2285
  - factor (option) ..... 7, 13, 31, 113, 179
  - fancyvrb (package) ..... 26
  - final (option) ..... 9, 31, 111, 112, 179
  - fix-cm (package) ..... 54
  - \font ..... 779, 1637
  - \font@name ..... 109, 817, 819, 821, 823, 835, 836, 1534, 1539, 1563, 1569, 1579, 1582, 1625, 1637, 1649, 1651, 1656, 1658, 1661, 1690, 1821, 1823, 2242, 2244, 2247, 2268, 2270, 2272, 2286, 2298, 2318
  - \fontcharwd ..... 1045
  - \fontdimen ..... 974, 981, 1066, 1071, 1170, 1173, 1552, 1553, 1569, 1649, 1683, 1684, 1691, 1692, 1698, 1702, 1709, 1719, 1723, 1726, 1834, 1837
  - fontinst (package) ..... 174
  - fontinstallationguide (package) ..... 152
  - french (package) ..... 180
  - frenchpro (package) ..... 84, 179, 183
- G**
- german (package) ..... 34, 178
  - graphics (package) ..... 9, 118
- H**
- hfoldsty (package) ..... 21, 126
  - hyperref (package) ..... 9, 50, 87, 118, 180, 181, 184
- I**
- IEEEtran (package) ..... 87
  - \iffontchar ..... 1076
  - \ifMT@auto ..... 216, 3658, 3705
  - \ifMT@babel ..... 216, 3824, 3871
  - \ifMT@do ..... 855, 919, 1514, 1853
  - \ifMT@document ..... 258, 3113
  - \ifMT@draft ..... 216, 3596
  - \ifMT@expansion ..... 216, 3626, 3702
  - \ifMT@if@ ..... 215, 680, 686, 704, 735, 2312, 3217, 3891, 3894
  - \ifMT@inannot ..... 98
  - \ifMT@inlist@ ..... 509, 554, 613, 637, 796, 886, 903, 912, 930, 948, 1512, 1904, 1915, 1964, 2012, 2306, 2370, 2382, 2686
  - \ifMT@kerning ..... 216, 3766, 3813, 3884
  - \ifMT@noligatures ..... 216, 3840
  - \ifMT@nonselected ..... 1259, 2025, 2033
  - \ifMT@noreset ..... 2087, 2094
  - \ifMT@opt@auto ..... 3224, 3661
  - \ifMT@opt@DVI ..... 3224, 3612
  - \ifMT@opt@expansion ..... 3224, 3607

- `\ifMT@protrusion` ..... 216, 643, 665, 3614, 3751  
`\ifMT@selected` ..... 216, 3688, 3708  
`\ifMT@spacing` ..... 216, 3757, 3807, 3823  
`\ifMT@tracking` ..... 216, 3741, 3801  
`ifpdf` (package) ..... 9, 118  
`\iftracingmicrotypeinpdfall` ..... 103  
`inputenc` (package) .....  
... 14, 27, 30, 51, 62, 82, 84, 86, 180, 182–184
- J**
- `jurabib` (package) ..... 87
- K**
- `kerning` (option) ..... 7, 11, 30, 111  
`keyval` (package) ..... 39, 58, 127, 183  
`\knaccode` ..... 1465, 1466, 1478, 1481, 1487  
`\knbccode` ..... 1458, 1459, 1477, 1480, 1486  
`\knbscode` ..... 1377, 1378, 1403, 1407, 1413
- L**
- `ledmac` (package) ..... 28, 31, 49, 87, 180  
`ledpar` (package) ..... 28  
`\leftmarginkern` ..... 651  
`letterspace` (option) . 9, 18, 24, 30, 42, 69, 113, 182  
`letterspace` (package) .....  
... 1, 24, 29, 30, 33, 37, 48, 70, 72, 182–184  
`\letterspacefont` ..... 1539  
`lineno` (package) ..... 49  
`listings` (package) ..... 26, 51, 62, 78, 181–183  
`lmodern` (package) ..... 8, 139  
`\LoadMicrotypeFile` ..... 22, 44, 73, 2682, 3601  
`\lpcode` ..... 987, 1019, 1020, 1206, 1209, 1591  
`\slig` ..... 24, 54, 1642, 1645  
`\lsstyle` ..... 23, 51,  
727, 750, 756, 1626, 1784, 1787, 1801, 3604
- M**
- `marvosym` (package) ..... 21, 30, 33, 131, 171, 182  
`mathdesign` (package) ..... 21, 127, 181  
`memoir` (package) ..... 23, 87  
`\Microtype@Hook` ..... 115, 3454  
`\microtypecontext` .....  
... 22, 47, 725, 2340, 2350, 3603, 3854, 3857  
`\microtypesetup` ..... 10, 46, 3458, 3602  
`miniltx` (package) ..... 1, 24, 29, 33, 37, 48, 184  
`minimal` (package) ..... 52  
`MT@efont` ..... 309, 799,  
815, 976, 1067, 1080, 1524, 1967, 2038, 2273  
`MT@abbr@ex` ..... 589  
`MT@abbr@ex@c` ..... 589  
`MT@abbr@ex@inh` ..... 589  
`MT@abbr@kn` ..... 589  
`MT@abbr@kn@c` ..... 589  
`MT@abbr@kn@inh` ..... 589  
`MT@abbr@nl` ..... 589  
`MT@abbr@pr` ..... 589  
`MT@abbr@pr@c` ..... 589  
`MT@abbr@pr@inh` ..... 589  
`MT@abbr@sp` ..... 589  
`MT@abbr@sp@c` ..... 589  
`MT@abbr@sp@inh` ..... 589  
`MT@abbr@tr` ..... 589  
`MT@abbr@tr@c` ..... 589  
`MT@active@features` .....  
825, 2300, 2303, 2315, 2322, 2359, 2369,  
2708, 3615, 3703, 3748, 3758, 3767, 3846, 3847  
`MT@addto@annot` ..... 98  
`MT@addto@setup` ..... 628, 630,  
689, 1183, 2236, 2340, 2341, 2905, 3328,  
3459, 3595, 3739, 3800, 3839, 3845, 3870, 3903  
`MT@auto` ..... 1268, 3657, 3671, 3681, 3685  
`MT@auto@` ..... 1268, 1277, 1321  
`MT@autofalse` ..... 218, 3669, 3680  
`MT@autotruer` ..... 218, 3418  
`MT@babelfalse` ..... 225  
`MT@babeltrue` ..... 225  
`MT@begin@catcodes` 1905, 1906, 1946, 2690, 2703,  
2728, 2742, 2764, 2783, 2797, 3000, 3431, 3432  
`MT@cat` ..... 1179, 2228, 2230  
`MT@cfg@catcodes` ..... 737, 1201, 1931, 1948  
`MT@char` ..... 1009,  
1019, 1020, 1025, 1026, 1029, 1031, 1045,  
1046, 1076, 1206–1210, 1305, 1306, 1308,  
1310, 1348–1350, 1377, 1378, 1384, 1385,  
1391, 1392, 1395, 1397, 1403–1408, 1458,  
1459, 1465, 1466, 1469, 1471, 1477–1481,  
1862, 1863, 2067, 2083, 2084, 2089, 2160,  
2163, 2165, 2191, 2194, 3060, 3061, 3066, 3069  
`MT@char@` .....  
2067, 2073, 2078, 2083, 2097, 2099, 2105,  
2106, 2108, 2121, 2122, 2125, 2126, 2129,  
2130, 2134, 2136, 2164, 2177, 2180, 2183, 2202  
`MT@charstring` ..... 2080, 2163, 2168  
`MT@check@active@set` .....  
3446, 3621, 3723, 3750, 3761, 3771  
`MT@check@font` ..... 795, 2297, 2361  
`MT@check@font@cx` ..... 2301, 2361  
`MT@check@range` ..... 3188, 3190  
`MT@check@range@` ..... 3190, 3191  
`MT@check@rlist` ..... 3141, 3181  
`MT@check@rlist@` ..... 3181, 3182  
`MT@check@step` ..... 3711  
`MT@checklist@` ..... 863, 879, 1850  
`MT@checklist@family` ..... 897  
`MT@checklist@font` ..... 941  
`MT@checklist@size` ..... 925  
`MT@checksetup` ..... 3468, 3479, 3486, 3505, 3524  
`MT@clear@options` ..... 187, 210, 295, 624, 3589  
`MT@clist@break` ..... 481, 892, 920, 936, 950  
`MT@clist@function` ..... 481  
`MT@cnt@encoding` ..... 3079, 3087, 3088  
`MT@cnt@family` ..... 3085, 3094, 3095  
`MT@cnt@series` ..... 3092, 3101, 3102  
`MT@cnt@shape` ..... 3099, 3107, 3108  
`MT@config@file` 3421, 3429, 3430, 3434, 3435, 3438  
`MT@context` ..... 1988, 2000, 2019, 2046  
`MT@copy@font` 784, 812, 2902, 2907, 3345, 3347, 3709  
`MT@copy@font@` ..... 812, 2902, 2907, 3345  
`MT@count` ..... 419, 420, 573, 1038, 1045, 1047,  
1048, 1053, 1058, 1059, 1063, 1071, 1102, 1164  
`MT@curr@file` ..... 1909, 1910,  
1919, 1920, 2692, 2693, 2876, 3022, 3434, 3905  
`MT@curr@list@name` ..... 1069, 1105, 1195,  
1317, 1890, 2057, 2197, 2205, 2210, 2218, 2224  
`MT@curr@ls` ..... 1560, 1646, 1649

- \MT@curr@set@name ..... 2421, 2423–2426, 2428, 2429, 2434, 2439, 2443, 2444, 2469, 2473, 2502, 2533, 2539, 2545, 2549, 2550, 2779, 2837, 2855, 2871, 2883, 2930, 2934, 2939, 2942, 2944, 2948, 2951, 2954, 2964, 2977, 2983, 2991, 3023, 3025  
 \MT@declare@char@inh ..... 3005, 3010, 3020  
 \MT@declare@sets ..... 2404, 2410, 2422, 2710  
 \MT@DeclareSet ..... 2399, 2401, 2418  
 \MT@DeclareSetAndUseIt ..... 2398, 2417  
 \MT@DeclareVariants ..... 2658, 2659, 2661  
 \MT@def@bool@opt ..... 3299, 3312–3314, 3323, 3335, 3343, 3352, 3359  
 \MT@def@n ..... 300, 2392, 2393  
 \MT@default@ex@set ..... 2643  
 \MT@default@kn@set ..... 2643  
 \MT@default@pr@set ..... 2643  
 \MT@default@sp@set ..... 2643  
 \MT@default@tr@set ..... 2643  
 \MT@define@code@key ..... 2819, 2863–2866, 3045–3047  
 \MT@define@code@key@font ..... 2842, 2868, 3049  
 \MT@define@code@key@size ..... 2830, 2867, 3048  
 \MT@define@opt@key ..... 2869, 2885–2888, 2987–2989  
 \MT@define@optionX ..... 3458, 3514, 3515  
 \MT@define@optionX@ ..... 3516, 3543, 3545, 3546  
 \MT@define@set@key@ ..... 2432, 2591–2594  
 \MT@define@set@key@font ..... 2537, 2596  
 \MT@define@set@key@size ..... 2462, 2595  
 \MT@detokenize@c ..... 339, 2075, 2161  
 \MT@detokenize@n ..... 339, 2193  
 \MT@dimen@six 973, 1039, 1063, 1102, 1718, 1721, 1830  
 \MT@dinfol ..... 86  
 \MT@dinfolist ..... 876, 887, 890, 895, 904, 907, 913, 915, 923, 931, 934, 939, 949, 952, 956  
 \MT@dinfoln ..... 86  
 \MT@DisableLigatures ..... 2701  
 \MT@do@font ..... 571, 989, 1281, 1416, 1488, 1591  
 \MT@documentfalse ..... 258  
 \MT@documenttrue ..... 258, 3849  
 \MT@dofalse ..... 855, 867, 891, 908, 935, 953  
 \MT@dottrue ..... 855, 858, 888, 905, 914, 932, 1845  
 \MT@draftfalse ..... 221, 3361  
 \MT@drafttrue ..... 221, 3363  
 \MT@edef@n ..... 302, 2024, 2050, 2386, 2825, 2876, 2878, 3021, 3036, 3168, 3392  
 \MT@encoding ..... 845, 945, 1983, 1995, 2075, 2161, 2192, 2209, 2216, 2224  
 \MT@encoding@check ..... 3034, 3038  
 \MT@end@catcodes ..... 1907, 1950, 2698, 2712, 2739, 2761, 2780, 2794, 2808, 3015, 3433  
 \MT@error ..... 75, 285, 616, 692, 1785, 1882, 1895, 2500, 2622, 2649, 2716, 2895, 2910, 2920, 3337, 3354, 3491, 3509, 3532, 3662, 3674, 3744, 3802, 3808, 3814, 3866  
 \MT@ex@c@name ..... 1252, 1254, 1323, 1324, 1334, 1340, 1341, 1353, 2745  
 \MT@ex@context ..... 815, 2353, 2391  
 \MT@ex@doc@contexts ..... 2391  
 \MT@ex@factor ..... 226, 1269, 1327  
 \MT@ex@factor@ ..... 1269, 1283, 1286, 1295, 1296, 1321  
 \MT@ex@inh@name ..... 1307, 1308, 1310  
 \MT@ex@level ..... 226, 3704, 3706  
 \MT@ex@max ..... 238, 1298, 1299  
 \MT@ex@min ..... 238, 1301, 1302  
 \MT@ex@setname ..... 2614  
 \MT@ex@split@val ..... 1293  
 \MT@exp@one@n ..... 310, 488, 612, 787, 789, 840, 883, 946, 1511, 1914, 2297, 2304, 2336, 2369, 2380, 2391, 2410, 2590, 2607, 2636, 2685, 3010, 3854  
 \MT@exp@two@c ..... 312, 330, 336, 346, 523, 786, 819, 821, 823, 834, 1625, 1658, 1661, 2074, 2079, 2162  
 \MT@exp@two@n ..... 314, 901, 910, 2438, 2544  
 \MT@expandfont ..... 1256, 1272, 1276  
 \MT@expansion ..... 801, 1244, 3733  
 \MT@expansionfalse ..... 217, 3608, 3698  
 \MT@expansiontrue ..... 217, 3417  
 \MT@extra@context ..... 2391, 2731, 2745, 2768, 2786, 2800, 2848, 2851, 2852, 2854, 2889, 2903, 2908, 2996, 3140, 3144, 3147, 3150, 3153, 3157, 3158, 3160, 3187  
 \MT@extra@inputenc ..... 2995, 3019, 3024, 3025  
 \MT@factor@default ..... 250, 3398, 3618  
 \MT@family ..... 787, 845, 902, 945, 1984, 2677, 2678  
 \MT@familyalias ..... 788, 789, 851, 909, 911, 1994, 1996, 2679  
 \MT@feat ..... 855, 978, 1011, 1084, 1086, 1088, 1089, 1091, 1092, 1098, 1100, 1103, 1111–1115, 1117, 1119–1123, 1126, 1127, 1132, 1136, 1139, 1142, 1145–1147, 1156, 1157, 1162, 1180, 1627, 1880, 1882, 1885, 1886, 1890, 1892, 1895, 2020, 2022, 2024, 2028, 2032, 2036, 2037, 2040, 2048, 2050, 2051, 2053, 2060, 2064, 2198, 2199, 2228, 2230  
 \MT@features ..... 609, 2391, 2404, 2590, 2601, 2630, 2873, 3005, 3442  
 \MT@features@long ..... 609, 612, 617, 2366, 3017  
 \MT@file@list ..... 1901, 1903, 1911, 1914, 1916, 1921, 1924, 2685, 2689  
 \MT@find@file ..... 787, 789, 1901  
 \MT@fix@catcode ..... 5  
 \MT@fix@font@set ..... 2457, 3613  
 \MT@font ..... 309, 786, 834, 841, 974, 981, 987, 988, 1019, 1020, 1025, 1026, 1045, 1046, 1066, 1071, 1076, 1170, 1173, 1206, 1207, 1209, 1210, 1277, 1281, 1305, 1306, 1348, 1350, 1377, 1378, 1384, 1385, 1391, 1392, 1403–1405, 1407, 1408, 1413–1415, 1458, 1459, 1465, 1466, 1477, 1478, 1480, 1481, 1486, 1487, 1511, 1515, 1559, 1854, 2232, 2268–2270, 2272, 2285, 2286, 2297, 2299, 2305, 2326, 2337  
 \MT@font@copy ..... 815, 816, 823, 824, 834–836  
 \MT@font@list ..... 2232, 2297–2299, 2360  
 \MT@font@orig ..... 816  
 \MT@font@sets ..... 2442, 2457, 2548, 3613  
 \MT@gdef@n ..... 300, 2724, 2738, 2760, 2793, 2807, 2944, 2964, 2977, 3027, 3444  
 \MT@get@axis ..... 2564–2567, 2577  
 \MT@get@basefamily ..... 1913, 1951  
 \MT@get@basefamily@ ..... 1957, 1960  
 \MT@get@char@unit ..... 1010, 1134, 1160, 1248  
 \MT@get@charwd ..... 1044, 1063, 1137, 1160  
 \MT@get@config ..... 3421  
 \MT@get@ex@opt ..... 1247, 1263, 1321

- \MT@get@ex@opt@ ..... 1329–1331, 1333, 1339  
 \MT@get@font ..... 2543, 2553  
 \MT@get@font@ ..... 2555, 2561, 2861  
 \MT@get@font@and@size ..... 2847, 2859  
 \MT@get@font@dimen ..... 1065, 1140  
 \MT@get@font@dimen@six . . . 961, 973, 1361, 1442, 1525  
 \MT@get@highlevel . . . 2437, 2447, 2579, 2824, 3035  
 \MT@get@inh@list ..... 964, 1250, 1364, 1445, 2046  
 \MT@get@listname ..... 1966, 2022, 2048  
 \MT@get@listname@ ..... 1966  
 \MT@get@ls@basefont ..... 1538, 1648, 1655  
 \MT@get@opt ..... 962, 1109, 1362, 1443  
 \MT@get@range ..... 2466, 2476, 2834  
 \MT@get@size ..... 2483, 2488, 2497, 2516, 2575  
 \MT@get@slot ..... 1008, 1861, 2067, 3059, 3065  
 \MT@get@space@unit .....  
 . . . 1134, 1234, 1375, 1382, 1389, 1456, 1463  
 \MT@get@tr@opt ..... 1527, 1596  
 \MT@get@tr@opt@ ..... 1614–1617, 1619  
 \MT@get@unit ..... 1142, 1150, 1607  
 \MT@get@unit@ ..... 1150  
 \MT@getkey ..... 3574, 3591  
 \MT@glot . . . 298, 629, 819, 1658, 1787, 2266, 2361–  
 2364, 2679, 2902, 3082, 3345, 3347, 3784, 3789  
 \MT@gobble@settings ..... 972, 979  
 \MT@if@false ..... 215,  
 676, 683, 700, 717, 2309, 3185, 3885, 3892  
 \MT@if@list@exists .....  
 . . . 960, 1246, 1262, 1360, 1441, 1526, 2019  
 \MT@if@true .....  
 215, 677–679, 684, 685, 701–703, 730, 734,  
 2302, 3194, 3198, 3206, 3211, 3886–3890, 3893  
 \MT@ifdefined@c@T .....  
 . . . 318, 680, 706, 1028, 1052, 1307, 1394,  
 1468, 1555, 2149, 2677, 3024, 3129, 3454, 3457  
 \MT@ifdefined@c@TF .....  
 . . . . . 318, 644, 1530, 1541, 1544, 1548,  
 1646, 1715, 1827, 1858, 1859, 2023, 2049, 3139  
 \MT@ifdefined@n@T .....  
 . . . . . 318, 634, 880, 898, 926, 942, 1029,  
 1145, 1181, 1308, 1334, 1395, 1469, 1599,  
 1601, 1620, 2002, 2227, 2424, 2672, 2748,  
 2879, 3088, 3095, 3102, 3108, 3140, 3150, 3574  
 \MT@ifdefined@n@TF ..... 318, 851,  
 861, 881, 899, 927, 943, 1111, 1119, 1323,  
 1340, 1848, 1885, 1990, 2075, 2616, 2619,  
 2645, 3115, 3167, 3176, 3240, 3443, 3447, 3852  
 \MT@ifdim . . . 428, 543, 544, 548, 549, 2499, 2508,  
 3192–3194, 3196, 3197, 3204–3206, 3209, 3210  
 \MT@ifdimen . . . . 393, 2528, 2933, 2941, 2953, 3407  
 \MT@ifempty ..... 348, 987, 988, 1017,  
 1023, 1223, 1224, 1374, 1381, 1388, 1413–  
 1415, 1427–1429, 1431–1433, 1455, 1462,  
 1486, 1487, 1503, 1504, 1696, 1697, 1707,  
 1725, 1797, 1832, 2371, 2403, 2408, 2449,  
 2455, 2477, 2478, 2491, 2492, 2554, 2569,  
 2580, 2600, 2605, 2629, 2634, 2707, 2775,  
 2860, 2870, 2875, 2889, 2901, 2908, 2962,  
 3004, 3009, 3039, 3253, 3284, 3423, 3465, 3521  
 \MT@ifint ..... 361, 2128, 2776, 2963, 3391, 3400  
 \MT@ifstreq ..... 435, 1187, 1881, 2021,  
 2036, 2178, 2181, 2335, 2542, 2846, 2932,  
 2950, 2952, 3256, 3258, 3261, 3264, 3286,  
 3288, 3302, 3303, 3369, 3370, 3375, 3379,  
 3406, 3467, 3474, 3478, 3485, 3523, 3529, 3826  
 \MT@in@clist ..... 509, 612, 635, 883, 901, 910,  
 946, 1511, 1903, 1914, 2297, 2304, 2369, 2685  
 \MT@in@rlist ..... 537, 929, 2011  
 \MT@in@rlist@ ..... 537  
 \MT@in@rlist@@ ..... 537  
 \MT@in@tlist ..... 525, 1963, 2380  
 \MT@in@tlist@ ..... 525  
 \MT@inannotfalse ..... 98  
 \MT@inannottrue ..... 98  
 \MT@increment ..... 573, 3087, 3094, 3101, 3107  
 \MT@info ..... 75, 90, 95, 3376,  
 3470, 3476, 3482, 3489, 3526, 3531, 3559, 3563  
 \MT@info@missing@char ..... 1048, 1074  
 \MT@info@nl ..... 75, 91, 92, 96, 645,  
 1075, 3367, 3377, 3430, 3448, 3451, 3611,  
 3617, 3624, 3705, 3734, 3749, 3755, 3760,  
 3764, 3770, 3774, 3805, 3811, 3817, 3862, 3873  
 \MT@inh@do ..... 2059, 3050  
 \MT@inh@feat ..... 2995, 3004, 3007, 3018  
 \MT@inh@split ..... 3052, 3056  
 \MT@inlist@false ..... 509, 513, 526, 538, 2312  
 \MT@inlist@true ..... 509, 515, 533, 545, 550, 2312  
 \MT@is@active ..... 2074, 2141  
 \MT@is@char ..... 2079, 2162, 2168  
 \MT@is@composite ..... 2077, 2189  
 \MT@is@feature ..... 611, 2409, 2606, 2635  
 \MT@is@letter ..... 2072, 2095, 2165, 2194  
 \MT@is@number ..... 2111, 2116  
 \MT@is@symbol ..... 2076, 2159  
 \MT@iterate ..... 559  
 \MT@kerning ..... 804, 1438, 3773  
 \MT@kerningfalse ..... 223  
 \MT@kerningtrue ..... 223  
 \MT@kn@c@name ..... 1447, 1449, 1494, 2800  
 \MT@kn@c@context ..... 2354, 2391  
 \MT@kn@doc@contexts ..... 2391  
 \MT@kn@factor ..... 226  
 \MT@kn@factor@ ..... 1111  
 \MT@kn@inh@name ..... 1468, 1469, 1471  
 \MT@kn@max ..... 238  
 \MT@kn@min ..... 238  
 \MT@kn@setname ..... 2614  
 \MT@kn@split@val ..... 1453  
 \MT@kn@unit ..... 226  
 \MT@kn@unit@ ..... 1119, 1497  
 \MT@led@kern ..... 641  
 \MT@led@unhbox@line ..... 641  
 \MT@ledmac@setup ..... 641, 698  
 \MT@let@cn ..... 307, 308, 852, 968, 1254,  
 1324, 1368, 1449, 1600, 1602, 1606, 1880,  
 1892, 2016, 2020, 2053, 2268, 2520, 2883, 3023  
 \MT@let@nc ..... 306,  
 640, 1147, 1690, 1821, 2032, 2060, 2064,  
 2286, 2307, 2360, 2426, 2428, 2434, 2539,  
 2753, 2930, 2934, 2939, 2942, 2948, 2951,  
 2954, 2983, 3169, 3174, 3262, 3265, 3481, 3488  
 \MT@let@nn ..... 308, 1112,  
 1117, 1120, 1132, 1341, 1344, 1621, 2379, 3450  
 \MT@letterspace .....  
 . . . 254, 1530, 1600, 1610, 3792, 3793, 3795

- `\MT@letterspace@` . . . . . 996, 1052, 1053, 1530–1532, 1535, 1539, 1543, 1552, 1560, 1569, 1589, 1591, 1592, 1721, 1725, 1796, 1833, 1836  
`\MT@letterspace@default` . . . . . 254, 3793  
`\MT@listname` . . . . . 1966, 2013, 2016, 2023, 2024, 2026, 2028, 2049, 2050, 2052, 2053, 2057, 2060, 3068, 3074  
`\MT@load@inputenc` . . . . . 1188, 1191, 1200  
`\MT@load@list` . . . . . 966, 1252, 1366, 1447, 1878  
`\MT@loop` . . . . . 559, 567, 3173  
`\MT@lower` . . . . . 2470, 2476, 2837  
`\MT@ls@adjust` . . . . . 1568, 1793, 1794  
`\MT@ls@basefont` . . . . . 1656–1658, 1661  
`\MT@ls@outer@k` . 1570, 1574, 1580, 1737, 1773, 1841  
`\MT@ls@too@large` . . . . . 1800, 1804, 3795  
`\MT@lsfont` . . . . . 1534, 1539, 1552, 1553, 1556, 1559, 1591, 1592, 1625, 1633, 1668, 1670, 1683, 1684, 1691, 1692, 1698, 1702, 1709, 1719, 1723, 1726, 1834, 1837  
`\MT@lslig` . . . . . 1645  
`\MT@lua` . . . . . 280, 366, 398, 440  
`\MT@map@clist@` . . . . . 481  
`\MT@map@clist@c` . . . 481, 825, 1860, 2303, 2315, 2322, 2359, 2366, 2404, 2406, 2601, 2603, 2630, 2632, 2873, 3005, 3007, 3017, 3442, 3572  
`\MT@map@clist@n` . . . . . 481, 860, 1847, 2391, 2435, 2464, 2540, 2590, 2662, 2722, 2822, 2832, 2844, 2946, 2960, 3062, 3248, 3251, 3280, 3282, 3311, 3386, 3463, 3519  
`\MT@map@tlist@` . . . . . 499  
`\MT@map@tlist@c` . . . . 499, 528, 539, 828, 1030, 1309, 1396, 1470, 1970, 2328, 3141, 3186, 3613  
`\MT@map@tlist@n` . . . . . 499, 1942, 2170, 3081  
`\MT@maybe@do` . . . . 855, 958, 1244, 1358, 1439, 1513  
`\MT@maybe@etex` . . . . . 259  
`\MT@maybe@rem@from@list` . . . . . 2330, 2334  
`\MT@EMT` . . . . . 2, 75, 78, 81, 83, 85, 90, 91, 93, 94, 199, 206, 287, 623, 693, 695, 2910, 3424, 3559, 3563, 3612, 3830, 3866, 3867  
`\MT@next@listname` . . . . . 1973, 1982  
`\MT@next@listname@` . . . . . 1991, 2003, 2009  
`\MT@n@ligatures` . . . . . 1854, 2701  
`\MT@n@setname` . . . . . 1846, 2701  
`\MT@n@ligatures` . . . . . 806, 1843, 3841  
`\MT@n@ligatures@` . . . . . 1668, 1670, 1854, 1857  
`\MT@n@ligatures@false` . . . . . 220  
`\MT@n@ligatures@true` . . . . . 220, 2709  
`\MT@n@selected@false` . . . . . 1259, 1273  
`\MT@n@selected@true` . . . . . 1259, 1261  
`\MT@n@rest@false` 2094, 2101, 2110, 2131, 2178, 2181  
`\MT@n@rest@true` . . . . . 2070, 2094, 2119  
`\MT@old@cmd` . . . . . 65, 70–73  
`\MT@opt@autofalse` . . . . . 3225  
`\MT@opt@autotru` . . . . . 3225, 3313  
`\MT@opt@def@set` . . . . . 3239, 3267, 3292  
`\MT@opt@DVIfalse` . . . . . 3226  
`\MT@opt@DVIttrue` . . . . . 3226, 3316, 3319  
`\MT@opt@expansion@false` . . . . . 3224  
`\MT@opt@expansion@true` . . . . . 3224  
`\MT@options` . . . . . 2810  
`\MT@optwarn@admissible` . . . . . 3227, 3304, 3379  
`\MT@optwarn@nan` . . . . . 3231, 3393, 3402  
`\MT@orig@add@accent` . . . . . 2287  
`\MT@orig@foreign@language` . . . . . 3879, 3881  
`\MT@orig@pickupfont` . . . . 723, 766, 771, 2237, 2289  
`\MT@orig@py@macron` . . . . . 763, 767, 772  
`\MT@orig@select@language` . . . . . 3874, 3876  
`\MT@outer@kern` . . . . . 1569, 1572, 1578, 1580, 1581, 1586, 1649, 1650, 1652, 1773, 1815, 1841  
`\MT@outer@space` . . . . . 1563, 1564, 1681, 1733, 1735, 1736, 1745, 1746, 1761, 1762, 1766, 1767  
`\MT@pdf@annot` . . . . . 98  
`\MT@pdf@tex@no` . . . . . 143, 193, 197, 272  
`\MT@permute` 2737, 2759, 2773, 2792, 2806, 3030, 3078  
`\MT@permute@` . . . . . 3078  
`\MT@permute@@` . . . . . 3078  
`\MT@permute@@@` . . . . . 3078  
`\MT@permute@@@@` . . . . . 3106, 3111  
`\MT@permute@@@@@` . . . . . 3119, 3121  
`\MT@permute@define` . . . . . 3112, 3122–3124, 3165  
`\MT@permute@reset` . . . . . 3081, 3171  
`\MT@permutelist` . . . . . 2731, 2745, 2771, 2786, 2800, 2848, 2849, 2854, 3028, 3135, 3136, 3140, 3144, 3147, 3150–3153, 3160, 3161, 3187, 3218, 3219  
`\MT@pickupfont` . . . . . 768, 773, 2283, 2291  
`\MT@plain` . . . . . 123, 188, 626, 719, 748, 1624, 2235, 3232, 3385, 3567, 3571  
`\MT@pr@c@name` . . . . . 966, 968, 1214, 2731  
`\MT@pr@context` . . . . . 2353, 2391  
`\MT@pr@doc@contexts` . . . . . 2391  
`\MT@pr@factor` . . . . . 226, 3401, 3618, 3619  
`\MT@pr@factor@` . . . . . 1111  
`\MT@pr@inh@name` . . . . . 1028, 1029, 1031  
`\MT@pr@level` . . . . . 226, 3616, 3617  
`\MT@pr@max` . . . . . 238  
`\MT@pr@min` . . . . . 238  
`\MT@pr@setname` . . . . . 2614  
`\MT@pr@split@val` . . . . . 1015  
`\MT@pr@unit` . . . . . 226, 3408, 3620  
`\MT@pr@unit@` . . . . . 1119, 1217  
`\MT@preset@aux` . . . . . 1219, 1221, 1223, 1224, 1227, 1499, 1501, 1503, 1504  
`\MT@preset@aux@factor` 1219, 1227, 1427–1429, 1499  
`\MT@preset@aux@space` . 1221, 1227, 1431–1433, 1501  
`\MT@preset@ex` . . . . . 1335, 1352  
`\MT@preset@kn` . . . . . 1492  
`\MT@preset@kn@` . . . . . 1492  
`\MT@preset@pr` . . . . . 1212  
`\MT@preset@pr@` . . . . . 1212  
`\MT@preset@sp` . . . . . 1420  
`\MT@preset@sp@` . . . . . 1420  
`\MT@ProcessOptionsWithKV` . . . . . 3567, 3592  
`\MT@protrusion` . . . . . 800, 958, 3623  
`\MT@protrusion@false` . . . . . 216  
`\MT@protrusion@true` . . . . . 216, 3414  
`\MT@rbba@expansion` . . . . . 604  
`\MT@rbba@kerning` . . . . . 604  
`\MT@rbba@protrusion` . . . . . 604  
`\MT@rbba@spacing` . . . . . 604  
`\MT@rbba@tracking` . . . . . 604  
`\MT@register@font` . . . . . 809, 2299, 2362  
`\MT@register@font@cx` . . . . . 2321, 2362  
`\MT@register@subst@font` . . . . . 2274, 2298, 2363  
`\MT@register@subst@font@cx` . . . . . 2314, 2363

- `\MT@rem@from@clist` ..... 520, 840, 2336, 2814  
`\MT@rem@from@list` ..... 830, 838  
`\MT@repeat` ..... 559, 569, 3179  
`\MT@requires@luatex` .....  
     ..... 275, 363, 395, 437, 2893, 3334, 3743  
`\MT@requires@pdftex` ..... 271,  
     361, 393, 435, 641, 783, 792, 803, 805,  
     813, 994, 1036, 1050, 1284, 1357, 1438,  
     1508, 1665, 1843, 2701, 2891, 2976, 2982,  
     3332, 3416, 3516, 3659, 3684, 3737, 3821, 3838  
`\MT@res@a` .... 458, 460, 511, 518, 521, 523, 527, 532  
`\MT@res@b` ..... 459, 460, 521–523, 531, 532  
`\MT@reset@context` ..... 2344, 2347, 2351, 2378  
`\MT@reset@context@` ..... 2351, 2378  
`\MT@reset@ef@codes` ..... 1249, 1271, 1284, 1336  
`\MT@reset@ef@codes@` ..... 1279, 1287, 1291  
`\MT@reset@kn@codes` ..... 1444, 1451, 1483  
`\MT@reset@kn@codes@` ..... 1483  
`\MT@reset@pr@codes` ..... 963, 970, 991  
`\MT@reset@pr@codes@` ..... 991  
`\MT@reset@sp@codes` ..... 1363, 1370, 1410  
`\MT@reset@sp@codes@` ..... 1410  
`\MT@restore@catcodes` ... 5, 7, 8, 210, 295, 624, 3907  
`\MT@restore@p@h` ..... 674, 686, 735  
`\MT@saved@setupfont` ..... 3557, 3564  
`\MT@scale` ..... 580, 1039, 1085, 1099, 1102, 1296  
`\MT@scale@factor` .... 1041, 1060, 1083, 1229, 1354  
`\MT@scale@to@em` .. 1018, 1024, 1036, 1057, 1235,  
     1376, 1383, 1390, 1457, 1464, 1609, 1716, 1828  
`\MT@selected@false` ..... 219  
`\MT@selected@true` ..... 219  
`\MT@series` ..... 845, 945, 1985, 1997  
`\MT@set@all@ex` ..... 1279, 1355  
`\MT@set@all@kn` ..... 1483, 1505  
`\MT@set@all@pr` ..... 984, 991, 1225  
`\MT@set@all@sp` ..... 1410, 1435  
`\MT@set@babel@context` .... 3851, 3877, 3882, 3896  
`\MT@set@codes` ... 969, 999, 1255, 1369, 1450, 1893  
`\MT@set@curr@ls` ..... 1560  
`\MT@set@curr@ok` ..... 1572, 1573, 1586, 1750  
`\MT@set@curr@os` ..... 1562, 1751  
`\MT@set@default@set` ..... 2630, 2636, 2643  
`\MT@set@ex@codes` ..... 1275, 3689, 3691  
`\MT@set@ex@codes@n` ..... 1259, 1275, 3691  
`\MT@set@ex@codes@s` ..... 1245, 3689  
`\MT@set@ex@heirs` ..... 1311, 1347  
`\MT@set@inh@list` ..... 3001, 3003  
`\MT@set@inputenc` 965, 1178, 1251, 1365, 1446, 2058  
`\MT@set@inputenc@` ..... 1181, 1183  
`\MT@set@kn@codes` ..... 1440  
`\MT@set@kn@heirs` ..... 1472, 1476  
`\MT@set@listname` .....  
     967, 1110, 1253, 1322, 1367, 1448, 1598, 2197  
`\MT@set@lsfont` ..... 1558, 1624, 1633  
`\MT@set@named@keys` 2733, 2747, 2769, 2788, 2802, 2810  
`\MT@set@pr@codes` ..... 959  
`\MT@set@pr@heirs` ..... 1032, 1205  
`\MT@set@sp@codes` ..... 1359  
`\MT@set@sp@heirs` ..... 1398, 1402  
`\MT@set@tr@codes` ..... 1522, 1628, 1638  
`\MT@SetExpansion` ..... 2743, 2745  
`\MT@SetExtraKerning` ..... 2798, 2800  
`\MT@SetExtraSpacing` ..... 2784, 2786  
`\MT@SetProtrusion` ..... 2729, 2731  
`\MT@SetTracking` ..... 2765, 2767  
`\MT@setup@` ..... 625, 628, 629, 3326, 3327  
`\MT@setup@contexts` ..... 2343, 2358  
`\MT@setupfont` .. 780, 2277, 3557, 3560, 3564, 3600  
`\MT@setupfont@hook` 675, 699, 705, 710, 735, 742, 781  
`\MT@shape` ..... 845, 945, 1986, 1998  
`\MT@shorthand@off` ..... 3860, 3891, 3894  
`\MT@show@pdfannot` ..... 104  
`\MT@shrink` ..... 226, 1266,  
     3630, 3631, 3634, 3635, 3638, 3642, 3694, 3707  
`\MT@shrink@` ..... 1266, 1277, 1321  
`\MT@shrink@default` ..... 251  
`\MT@size` ..... 544, 548, 549, 845, 945  
`\MT@size@name` ..... 537, 2013  
`\MT@sp@c@name` ..... 1366, 1368, 1422, 2786  
`\MT@sp@context` ..... 2354, 2391, 3826  
`\MT@sp@doc@contexts` ..... 2391  
`\MT@sp@factor` ..... 226  
`\MT@sp@factor@` ..... 1111  
`\MT@sp@inh@name` ..... 1394, 1395, 1397  
`\MT@sp@max` ..... 238  
`\MT@sp@min` ..... 238  
`\MT@sp@setname` ..... 2614  
`\MT@sp@split@val` ..... 1372  
`\MT@sp@unit` ..... 226  
`\MT@sp@unit@` ..... 1119, 1425  
`\MT@spacing` ..... 804, 1357, 3763  
`\MT@spacing@false` ..... 222  
`\MT@spacing@true` ..... 222  
`\MT@split@codes` ..... 1001, 1005  
`\MT@split@name` ..... 786, 845  
`\MT@step` ..... 226,  
     1267, 3633, 3649, 3656, 3708, 3713, 3714, 3718  
`\MT@step@` ..... 1267, 1277, 1321  
`\MT@step@default` ..... 251  
`\MT@stretch` ..... 226, 1265, 3627,  
     3628, 3631, 3634, 3636, 3641, 3644, 3693, 3707  
`\MT@stretch@` ..... 1265, 1277, 1321  
`\MT@stretch@default` ..... 251, 3628  
`\MT@temp` ..... 986–989,  
     1412–1416, 1485–1488, 1686, 1690, 1693,  
     1698, 1701, 1706, 1745, 1752, 1755, 1758,  
     1761, 1766, 1773, 1776, 1818, 1821, 1838,  
     2563, 2588, 3570, 3575, 3580, 3581, 3584, 3588  
`\MT@tempencoding` ..... 3114, 3115, 3125, 3132  
`\MT@tempfamily` ..... 3126  
`\MT@tempseries` ..... 3127  
`\MT@tempshape` ..... 3128  
`\MT@tempsize` ..... 3082, 3129, 3139, 3141, 3145  
`\MT@test@east` ..... 1714, 1826, 2448, 2453  
`\MT@textls` ..... 49, 1793, 1794, 1796  
`\MT@the@pr@code` ..... 993, 1019, 1025, 1593  
`\MT@the@pr@code@tr` ..... 993, 1593  
`\MT@tlist@break` ..... 499, 534, 556, 1975, 3221  
`\MT@toks` ..... 214, 1075, 1105, 1317,  
     2071, 2099, 2106, 2153, 2183, 2202, 2214, 2222  
`\MT@tr@c@name` .....  
     ... 1599–1602, 1606, 1620, 1621, 2770, 2777  
`\MT@tr@context` ..... 2354, 2391  
`\MT@tr@doc@contexts` ..... 2391  
`\MT@tr@factor@` ..... 1608  
`\MT@tr@font@list` ..... 1509

- `\MT@tr@ispace` ..... 1541, 1542, [1614](#)  
`\MT@tr@ligatures` ..... 1555, [1617](#), 1667, 1670  
`\MT@tr@max` ..... [238](#), 1805–1807  
`\MT@tr@min` ..... [238](#), 1809–1811  
`\MT@tr@noligatures` ..... 1555, [1664](#)  
`\MT@tr@okern` ..... 1548, [1616](#)  
`\MT@tr@ospace` ..... 1544, 1545, [1614](#)  
`\MT@tr@outer@icr` ..... 1755, [1778](#)  
`\MT@tr@outer@icr@` ..... [1778](#)  
`\MT@tr@outer@l` ..... 1566, [1730](#)  
`\MT@tr@outer@next` ..... [1741](#), 1780  
`\MT@tr@outer@r` ..... 1588, [1741](#), 1778  
`\MT@tr@outer@r@` ..... [1741](#)  
`\MT@tr@set@okern` ..... 1549, [1815](#)  
`\MT@tr@set@okern@` ..... 1819, 1820, [1825](#)  
`\MT@tr@set@space` ..... 1547, [1682](#)  
`\MT@tr@set@space@` ..... 1687–1689, [1695](#)  
`\MT@tr@set@space@@` ..... 1700, 1705, 1708, [1713](#)  
`\MT@tr@set@name` ..... [2614](#)  
`\MT@tr@unit@` ..... [1601](#), 1715, 1827  
`\MT@tracking` .....  
     793, [1509](#), 1628, 2279, 3543, 3544, 3745, 3754  
`\MT@tracking@` ..... [1509](#), 3544  
`\MT@trackingfalse` ..... [224](#)  
`\MT@trackingtrue` ..... [224](#)  
`\MT@try@order` ..... 1970, [1978](#)  
`\MT@undefined@char` ..... 2146, 2148, [2158](#)  
`\MT@upper` ..... 2470, [2476](#), 2837  
`\MT@use@set` ..... 2601, 2607, [2614](#)  
`\MT@eval` ..... 2371–2373, 2375, 2381, 2383,  
     2386, 2436, 2440, 2448, 2450, 2465–2467,  
     2479, 2482, 2484, 2487, 2489, 2490, 2496,  
     2498, 2499, 2501, 2504, 2506, 2517, 2520,  
     2524, 2528–2530, 2532, 2534, 2541–2543,  
     2546, 2556–2558, 2571, 2573, 2578, 2580,  
     2582, 2584, 2823, 2825, 2833–2835, 2845–  
     2847, 2850, 2855, 3033, 3034, 3036, 3040,  
     3061, 3068, 3073, 3074, 3240, 3241, 3244,  
     3252, 3253, 3256, 3258, 3261, 3264, 3283,  
     3284, 3286, 3288, 3464, 3465, 3467, 3474,  
     3478, 3485, 3491, 3520, 3521, 3523, 3529, 3532  
`\MT@variants` ..... 1963, [2655](#), [2665](#)  
`\MT@vinfo` ..... 75, 92, 799, 873, 1114, 1123,  
     1127, 1162, 1325, 1342, 1524, 1532, 1865,  
     1874, 1886, 1910, 1920, 1923, 2026, 2028,  
     2034, 2352, 2375, 2687, 2693, 3367, 3374, 3853  
`\MT@warn@ascii` ..... 2108, [2201](#)  
`\MT@warn@axis@empty` ..... 2570, 2581, [2586](#)  
`\MT@warn@code@too@large` ..... 1089, 1092, [1096](#)  
`\MT@warn@err` ..... 75, 3371, 3372  
`\MT@warn@ex@too@large` ..... 1299, 1302, [1315](#)  
`\MT@warn@maybe@inputenc` ..... 2215, 2223, [2226](#)  
`\MT@warn@nodim` ..... [2925](#), 2935, 2943, [2955](#)  
`\MT@warn@number@too@large` ..... 2135, [2207](#)  
`\MT@warn@preset@t@width` ... 1218, [1238](#), 1426, 1498  
`\MT@warn@rest` ..... 2088, [2212](#)  
`\MT@warn@tracking@DVI` ..... 1533, [3777](#)  
`\MT@warn@unknown` ..... 2085, [2220](#)  
`\MT@warning` ..... 66, 75, 93, 1806,  
     1810, 1868, 2037, 2425, 2532, 2587, 2673,  
     2695, 2778, 2880, 2926, 2966, 2979, 3041,  
     3133, 3151, 3218, 3371, 3376, 3454, 3549–3551  
`\MT@warning@nl` [75](#), 87, 94, 196, 286, 622, 657, 666,  
     712, 975, 1067, 1103, 1155, 1195, 1239,  
     1316, 1675, 2202, 2208, 2213, 2221, 2250,  
     2750, 3228, 3234, 3244, 3372, 3377, 3409,  
     3437, 3597, 3651, 3695, 3716, 3779, 3827, 3898  
`\MT@while@num` ..... [565](#), 571, 738–740  
`\MT@with@babel@and@T` .....  
     ... [633](#), 677–679, 701–703, 3886–3890, 3893  
`\MT@with@package@T` ..... 631, 681, 684, 685,  
     691, 698, 708, 720, 734, 736, 749, 762, 2261  
`\MT@xadd` ..... 102, [467](#), 1911, 1916, 1921,  
     1924, 2316, 2324, [2383](#), 2468, 2689, 2836, 3067  
`\MT@xaddb` ..... [474](#), 2853, 3143  
`\MT@xdef@n` .. 302, 2617, 2620, 2647, [2652](#), 2676,  
     2777, 2848, 2871, 2991, 3025, 3160, 3241, 3243
- N**
- `\normalfont` ..... 779, 3726, 3730  
`\nullfont` ..... 779
- P**
- `\pdfadjustinterwordglue` ..... 3545, 3759  
`\pdfadjustspacing` ..... 3515, 3704, 3729  
`\pdfappendkern` ..... 3547, 3769  
`\pdfcopyfont` ..... 823  
`pdfcpot` (package) ..... [4](#), [26](#), [29](#), [50](#), [59](#), [86](#), [178](#)  
`\pdffontexpand` ..... 1277  
`\pdfnoligatures` ..... 1867, 1873  
`\pdfprependkern` ..... 3546, 3768  
`\pdfprotrudechars` ..... 3514, 3616, 3751, 3788  
`\pdftracingfonts` ..... 274  
`\pickup@font` ..... 723, 766, 768,  
     771, 773, 2249, 2251, [2260](#), 2283, 2289, 2291  
`pifont` (package) ..... [87](#)  
`pinyin` (package) ..... [51](#), [183](#)  
`protrusion` (option) ..... [6](#), [7](#), [11](#), [110](#), [111](#)  
`PSNFSS` (package) ..... [174](#)  
`pstricks` (package) ..... [9](#), [118](#)  
`pxfonts` (package) ..... [21](#), [126](#)
- Q**
- `qfonts` (package) ..... [21](#), [126](#), [181](#)
- R**
- `relsize` (package) ..... [94](#)  
`\rightmarginkern` ..... 650  
`\rpostcode` ..... 988, 1025, 1026, 1207, 1210, 1592
- S**
- `selected` (option) ..... [8](#), [14](#), [31](#), [63](#), [81](#), [111](#), [179](#)  
`\selectfont` ..... 1629, 2346, 2356, 3904  
`\set@fontsize` ..... 2524  
`\SetExpansion` ..... [14](#), 36, [2741](#)  
`\SetExtraKerning` ..... [19](#), 38, [2796](#)  
`\SetExtraSpacing` ..... [18](#), 39, [2782](#)  
`\SetProtrusion` ..... [13](#), 35, [2727](#)  
`\SetTracking` ..... [16](#), 37, [2763](#)  
`\sfcode` ..... 3825  
`\shbcode` ..... 1391, 1392, 1405, 1408, 1415  
`\showhyphens` ..... 3724, [3728](#)  
`shrink` (option) .. [8](#), [16](#), [31](#), [113](#), [119](#), [121](#), [181](#), [183](#)  
`slantsc` (package) ..... [152](#)  
`soul` (package) ..... [4](#), [29](#), [51](#), [74](#), [182](#), [184](#)

soulutf8 (package) . . . . .	29, 184	txfonts (package) . . . . .	21, 126
spacing (option) . . . . .	7, 11, 30, 111		
\stbrcode . . . . .	1384, 1385, 1404, 1408, 1414		
step (option) . . . . .	8, 31, 113, 119, 121, 179, 181, 183		
stretch (option) . . . . .	8, 16, 25, 31, 113, 119, 121, 183		
	<b>T</b>		<b>U</b>
\tagcode . . . . .	1556, 1859, 1863	ucs (package) . . . . .	84, 184
tex4ht (package) . . . . .	50, 182	ulgothic (package) . . . . .	21
\textls . . . . .	23, 52, 53, 728, 751, 757, 1641, 1791	unit (option) . . . . .	8, 31, 180–182
\textls* . . . . .	23	\UseMicrotypeSet . . . . .	11, 33, 60, 72, 2419, 2598
\textmicrotypecontext . . . . .	22, 48, 724, 2350		
tipa (package) . . . . .	87		<b>V</b>
trace (package) . . . . .	88, 182	verbose (option) . . . . .	9, 29, 31, 113, 180, 184
\tracingmicrotype . . . . .	86		
\tracingmicrotypeinpdf . . . . .	97		<b>W</b>
\tracingmicrotypeinpdfallfalse . . . . .	103	wordcount (package) . . . . .	48, 183
\tracingmicrotypeinpdfalltrue . . . . .	103		
tracking (option) . . . . .	7, 11, 23, 30, 111		<b>Y</b>
			<b>Z</b>
		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 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.