

# The microtype package

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

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

v2.2  
2007/07/14

---

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

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

---

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

## Contents

<b>1</b>	<b>Micro-Typography with pdf<math>\TeX</math></b>	<b>4</b>
<b>2</b>	<b>Getting Started</b>	<b>5</b>
<b>3</b>	<b>Options</b>	<b>6</b>
3.1	Micro-Typographic Options . . . . .	6
3.2	Options for Character Protrusion . . . . .	7
3.3	Options for Font Expansion . . . . .	7
3.4	Option for Tracking/Letterspacing . . . . .	8
3.5	Miscellaneous Options . . . . .	8
3.6	Changing Options Later . . . . .	9
<b>4</b>	<b>Selecting Fonts for Micro-Typography</b>	<b>10</b>
<b>5</b>	<b>Micro Fine Tuning</b>	<b>12</b>
5.1	Character Protrusion . . . . .	12
5.2	Font Expansion . . . . .	14
5.3	Tracking . . . . .	16
5.4	Interword Spacing . . . . .	17
5.5	Additional Kerning . . . . .	18
5.6	Character Inheritance . . . . .	19
5.7	Configuration Files . . . . .	19
<b>6</b>	<b>Context-sensitive Setup</b>	<b>21</b>
<b>7</b>	<b>Letterspacing revisited</b>	<b>22</b>
<b>8</b>	<b>Disabling Ligatures</b>	<b>23</b>
<b>9</b>	<b>Hints and Caveats</b>	<b>24</b>
<b>10</b>	<b>Contributions</b>	<b>26</b>
<b>11</b>	<b>Acknowledgments</b>	<b>26</b>
<b>12</b>	<b>References</b>	<b>27</b>
<b>13</b>	<b>Short History</b>	<b>28</b>
<b>14</b>	<b>Implementation</b>	<b>31</b>
14.1	Preliminaries . . . . .	32
	[Debugging – 33] [Requirements – 34] [Declarations – 36] [Auxiliary Macros – 38] [Compatibility – 45]	
14.2	Font Setup . . . . .	49
	[Protrusion – 54] [Expansion – 61] [Interword Spacing (Glue) – 63] [Additional Kerning – 65] [Tracking – 67] [Disabling Ligatures – 73] [Loading the Configuration – 73] [Translating Characters into Slots – 78] [Hook into $\LaTeX$ 's font selection – 83] [Context-sensitive Setup – 85]	

14.3	Configuration . . . . .	87
	[Font Sets – 87] [Variants and Aliases – 93] [Disabling Ligatures – 94]	
	[Interaction with babel – 94] [Fine Tuning – 95] [Character	
	Inheritance – 100] [Permutation – 102]	
14.4	Package Options . . . . .	105
	[Declaring the Options – 105] [Reading the Configuration File – 109]	
	[Hook for Other Packages – 110] [Changing Options Later – 111]	
	[Processing the Options – 113]	
<b>15</b>	<b>Configuration Files</b>	<b>120</b>
15.1	Font Sets . . . . .	120
15.2	Font Variants and Aliases . . . . .	121
15.3	Interaction with babel . . . . .	122
15.4	Note on Admissible Characters . . . . .	122
15.5	Character Inheritance . . . . .	123
	[OT1 – 123] [T1 – 123] [LY1 – 124] [OT4 – 125] [QX – 125] [T5 – 126]	
	[Euro symbols – 126]	
15.6	Tracking . . . . .	127
15.7	Font Expansion . . . . .	127
15.8	Character Protrusion . . . . .	128
	[Default – 129] [Italics – 136] [Small Caps – 145] [Italic Small Caps – 147]	
	[textcomp – 149] [Computer Modern math – 154] [AMS symbols – 158]	
	[Euler – 161] [Euro symbols – 166]	
15.9	Interword Spacing . . . . .	167
	[Nonfrenchspacing – 168]	
15.10	Additional Kerning . . . . .	169
	[French – 170] [Turkish – 170]	
<b>16</b>	<b>Auxiliary File for Micro Fine Tuning</b>	<b>171</b>
<b>A</b>	<b>Change History</b>	<b>173</b>
<b>B</b>	<b>Index</b>	<b>179</b>
<b>C</b>	<b>The L<sup>A</sup>T<sub>E</sub>X Project Public License</b>	<b>186</b>

## List of Tables

1	Predefined font sets . . . . .	11
2	Fonts with tailored protrusion settings . . . . .	20
3	Order for matching font attributes . . . . .	76

# 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 doesn't pose a problem any longer. The microtype

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

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

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

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

## 2 Getting Started

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

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

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

- Enable the respective micro-typographic feature, either via the respective package option or with the `\microtypesetup` command (section 3).
- Select the fonts to which this feature should be applied by declaring and activating ‘sets of fonts’. Some sets are predefined, which may be activated directly in the package options (section 4).
- Fine-tune the micro-typographic settings of the fonts or sets of fonts (section 5).
- If you’re of the kind who always wants to go further, you’ll certainly be interested in the possibility of context-sensitive setup (section 6).
- 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).

### 3.1 Micro-Typographic Options

**protrusion** `true, false, compatibility, nocompatibility, <font set name>` \*true

**expansion** These are the main options to control the level of micro-typographic refinement which the fonts in your document should gain. By default, the package is moderately greedy: character protrusion will be enabled, font expansion will only be disabled in circumstances where pdfT<sub>E</sub>X cannot expand the fonts automatically, that is, if it is either too old (versions before 1.20) or if the output mode is DVI (see section 3.5). In other words, `microtype` will try to apply as much micro-typography as can safely be expected to work under the respective conditions (and it is usually not necessary to load the package with different options for PDF resp. DVI mode).

**activate** Protrusion and expansion may be enabled or disabled independently from each other by setting the respective key to `true` resp. `false`. The `activate` option is a shortcut for setting both options at the same time. Therefore, the following lines all have the same effect (when creating PDF files with a recent version of pdfT<sub>E</sub>X):

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

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

```
\usepackage{microtype}
```

When pdfT<sub>E</sub>X employs font expansion and character protrusion, line breaks (and consequently, page breaks) may turn out differently. If that is not desired, you

may pass the value `compatibility` to the protrusion and/or expansion options. Typographically, however, the results may be suboptimal, hence the default value is `nocompatibility`.

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

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

**spacing** The new extensions of tracking, interword spacing, and additional kerning do not

**kerning** have a compatibility level. Therefore, they can only be switched on or off, or they may be activated by passing a set name to the option. By default, neither feature is enabled.

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

### 3.2 Options for Character Protrusion

**factor** *<integer>* 1000

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

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

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

### 3.3 Options for Font Expansion

**auto** true, false \*true

As noted in chapter 1, the expanded versions of the fonts may be calculated automatically. This option is true by default provided that pdfTeX's version is found to be 1.20 or higher and the output mode is PDF; otherwise, it will be disabled. If auto is set to false, the fonts for all expansion steps must exist (with files called *<font name>±<expansion value>*, e. g., `cmr12+10`, as described in the [pdfTeX manual](#)).

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

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

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

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

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

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

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

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

```
\usepackage{microtype}
```

**selected** true, false false

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

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

### 3.4 Option for Tracking/Letterspacing

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

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

### 3.5 Miscellaneous Options

**DVIoutput** true, false false

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

Note that this will confuse packages that depend on the value of \pdfoutput if they were loaded earlier, as they had been made believe that they were called to generate PDF output where they actually weren’t. These packages are, among

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



others: graphics, color, hyperref, pstricks and, obviously, ifpdf. Either load these packages after microtype or else issue the command `\pdfoutput=0` earlier – in the latter case, the `DVIoutput` option is redundant.

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

**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 false

Information on the settings used for each font will be written into the log file if you enable the verbose option, which is disabled by default.

When microtype encounters a problem that is not fatal (e. g., an unknown character in the settings, or non-existent settings), it will by default only issue a warning and try to continue. Loading the package with verbose=errors will turn all warnings into errors, so that you can be sure that no problem will go unnoticed.

**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.

### 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 incompatibility (but not the name of a font set). Since there is no compatibility level for **tracking**, **spacing** and **kerning**, only the values true and false are admissible here. Using this command, you could for instance temporarily disable font expansion by saying:

```
\microtypesetup{expansion=false}
```

## 4 Selecting Fonts for Micro-Typography

By default, character protrusion will be applied to all text fonts that are being used in the document, and a basic set of fonts will be subject to font expansion. You may want to customise which fonts should get the benefit of micro-typographic treatment. This can be achieved by declaring and activating ‘font sets’; these font sets are specified via font attributes that have to match.

`\DeclareMicrotypeSet` [*<features>*] {*<set name>*} {*<set of fonts>*}

`\DeclareMicrotypeSet*` This command declares a new set of fonts to which the micro-typographic extensions should be applied. The optional argument may contain a comma-separated list of features to which this set should be restricted. The starred version of the command declares *and* activates the font set at the same time.

The *set of fonts* is specified by assigning values to the NFSS font attributes: encoding, family, series, shape and size (cf. [L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> font selection](#)). Let’s start with an example. This package defines a font set called ‘`basictext`’ in the main configuration file as follows:

```
\DeclareMicrotypeSet{basictext}
{ encoding = {OT1,T1,LY1,OT4,QX,T5},
  family   = {rm*,sf*},
  series    = {md*},
  size      = {normalsize,footnotesize,small,large}
}
```

If you now call

```
\UseMicrotypeSet[protrusion]{basictext}
```

in the document’s preamble, only fonts in the text encodings OT1, T1, LY1, OT4, QX or T5, roman or sans serif families, normal (or ‘medium’) series, and in sizes called by `\normalsize`, `\footnotesize`, `\small` or `\large`, will be protruded. Math fonts, on the other hand, will not, since they are in another encoding. Neither will fonts in bold face, or huge fonts. Etc.

If an attribute list is empty or missing – like the ‘shape’ attribute in the above example –, it does not constitute a restriction. In other words, this is equivalent to specifying *all* possible values for that attribute. Therefore, the predefined set ‘`alltext`’, which is declared as:

```
\DeclareMicrotypeSet{alltext}
{ encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
```

is far less restrictive. The only condition is that the encoding must match.

If a value is followed by an asterisk (like ‘`rm*`’ and ‘`sf*`’ in the above example), it does not designate an NFSS code, but will expand to the document’s `\<value>default`, e.g., `\rmdefault`. A single asterisk means `\<attribute>default`, e.g., `\encodingdefault`, respectively `\normalsize` for the size axis.

Sizes may be either 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`’);

Table 1: Predefined font sets

Set name	Font attributes				
	Encoding	Family	Series	Shape	Size
all	–	–	–	–	–
alltext (allmath)	Text encodings, TS1 (OML, OMS, U)	–	–	–	–
basictext (basicmath)	Text encodings (OML, OMS)	$\rm*$ , $\sf*$	$\md*$	–	$\normalsize$ , $\footnotesize$ , $\small$ , $\large$
smallcaps	Text encodings	–	–	sc	–
footnotesize	Text encodings, TS1	–	–	–	$\small$
scriptsize	Text encodings, TS1	–	–	–	$\footnotesize$
normalfont	$\encoding*$	$\family*$	$\series*$	$\shape*$	$\normalsize$

‘Text encodings’ = OT1, T1, LY1, OT4, QX, T5

‘...\*’ = ‘...default’

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. Size selection commands are possible, too, however, ranges are not allowed.

Table 1 lists the nine predefined font sets. They may also be activated by passing their name to the feature options protrusion, expansion, tracking, spacing and kerning when loading the package, for example:

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

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

This command activates a font set previously declared by `\DeclareMicrotypeSet`. Using the optional argument, you can limit the application of the set to one or

more features. This command only has an effect if the feature has been activated in the package options.

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

If a feature is enabled but no font set has been chosen explicitly, the sets declared by this command will be activated. By default, the ‘alltext’ font set will be used for character protrusion and additional kerning, the ‘basictext’ set for font expansion and interword spacing, and the ‘smallcaps’ set for tracking.

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

## 5 Micro Fine Tuning

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

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

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

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

### 5.1 Character Protrusion

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

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

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

<sup>4</sup> For the interested, table 3 on page 76 presents the exact order.

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

*The protrusion settings* consist of  $\langle \text{character} \rangle = \langle \text{protrusion factors} \rangle$  pairs.

The characters may be specified either as a single character (‘A’), as a text symbol command (‘\textquoteleft’), or as a slot number: three digits for decimal notation, prefixed with ‘#’ for hexadecimal, with ‘o’ for octal (e. g., the ‘fl’ ligature in T1 encoding: 029, #1D, o35). 8-bit (and even UTF-8) characters may be entered directly or in L<sup>A</sup>T<sub>E</sub>X’s traditional 7-bit notation: both ‘\A’ and ‘Ä’ 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`, with the exception of `uft8x`.

**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 has been loaded with the selected option. Otherwise, the expansion settings will be ignored.

The *expansion settings* consist of *<character> = <expansion factor>* pairs. You may specify one number for each character, which determines the amount that a character may be expanded. The numbers denominate thousandths of the full expansion. For example, if you set the expansion factor for the character 'O' to 500, it will only be expanded or shrunk by one half of the amount that the rest of the characters will be expanded or shrunk. While the default value for character protrusion is 0 – that is, if you didn't specify any characters, none would be protruded –, the default value for expansion is 1000, which means that all characters would be expanded by the same amount.

*Options:*

**name, load, preset, inputenc, context** Analogous to `\SetProtrusion`, the optional argument may be used to assign a name to the list, to load another list, to preset all expansion factors, to set the input encoding, or to determine the context of the list (contexts are only allowed if you are using at least pdfTeX version 1.40.4).

<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.

**auto**, **stretch**, **shrink**, **step** These keys can be used to override the global settings from the package options (see section 3.3). If you don't specify either one of stretch, shrink and step, their respective global value will be used (that is, no calculation will take place).

As a practical example, suppose you have a paragraph containing a widow that could easily be avoided by shrinking the font a little bit more. In conjunction with the context option (see section 6 for further details), you could thus allow for more expansion in this particular paragraph:

```
\SetExpansion
[ context = sloppy,
  stretch = 30,
  shrink   = 60,
  step     = 5 ]
{ encoding = {OT1,T1,TS1} }
{ }
% ... END PREAMBLE
{\microtypecontext{expansion=sloppy}%
  This paragraph contains an 'unnecessary' widow.}
```

This method of employing contexts to temporarily apply different expansion parameters only works with pdfTeX version 1.40.4 or later. For older versions, a dirty trick is laid out in section 14.2 on page 51. Also note that pdfTeX prohibits the use of fonts with different expansion limits or steps (even of different fonts) within one paragraph, hence the sloppy context has to be applied to complete paragraphs.

**factor** This option provides a different method to alter expansion settings for certain fonts, working around the restriction just mentioned. The factor option influences the expansion factors of all characters (in contrast to the overall stretchability) of the font. For instance, if you want the italic shape to be expanded less, you could declare:

```
\SetExpansion
[ factor = 500 ]
{ encoding = *,
  shape    = it }
{ }
```

The factor option can only be used to *decrease* the stretchability of the characters, that is, it may only receive values smaller than 1000. Also, it can only be used for single fonts or font sets; setting it globally in the package options wouldn't make much sense – to this end, you use the package's stretch and shrink options.

These options in the optional first argument will even be taken into account if the package has not been loaded with the selected option.

If the selected option has been passed to the package (cf. section 3.3), 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.

### 5.3 Tracking

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

An important typographic technique – which was missing in  $\TeX$  for a long time – is 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-*character* spacing gets 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, and the shrink amount will not be modified. If you don't specify the spacing option, the interword space will be scaled by the current letterspace amount (as in the above example), stretch and shrink will not be changed.

**outer spacing** If an interword space precedes or follows letterspaced 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.

**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,

<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.

<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 recommended to use at least version 1.40.4.



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).

As an example, suppose you want to space out all small capitals by 50/1000em (0.025em on each side of each character), 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:  $2 \times \backslash fontdimen 2$ ), as well as the maximum stretch and the maximum shrink amount of the interword space ( $\backslash fontdimen 3$  and  $4$ ). As another example, setting all three value to  $-1000$  would completely cancel a space after the respective character.

## 5.5 Additional Kerning

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

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

I should not neglect to mention a limitation of this additional kerning: words *immediately following* such a kern (not separated by a space) will not be hyphenated, unless you insert the breakpoints manually, e. g., for kerning after the apostrophe, ‘l ’apos\ -trophe’. This restriction of 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 are relative to 1em.

**context** When it comes to kerning settings, this option is especially useful, since it allows to apply settings depending on the current language (see section 6).

## 5.6 Character Inheritance

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

In most cases, accented characters should inherit the settings from the respective base character. For example, all of the characters Å, Á, Â, Ã, Ä, Å and Æ should probably be protruded by the same (absolute) amount as the character A. Using the command `\DeclareCharacterInheritance`, you may declare such classes of characters, so that you then only have to set up the respective base character. With the optional argument, which may contain a comma-separated list of features, you can confine the scope of the list. Additionally, it accepts the `inputenc` key to set the input encoding for this list. The font set can be declared in the usual way, with the only exception that exactly one encoding must be specified. The inheritance lists are declared as pairs of *<base character>* = *<list of inheriting characters>*. Unless you are using a different encoding or a very peculiarly shaped font, there should be no need to change the default character inheritance settings.

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

## 5.7 Configuration Files

The default configuration, consisting of inheritance settings, declarations of font sets and alias fonts, and generic protrusion, expansion, spacing and kerning settings, will be loaded from the file `microtype.cfg`. You may extend this file with custom settings (or load a different configuration file with the ‘`config`’ option, see section 3.5).

If you are embarking on creating new settings for a font family, you should put them into a separate file, whose name must be: ‘`mt-<font family>.cfg`’ (e. g., ‘`mt-cmr.cfg`’), and may contain all commands described in the current section 5. These files will be loaded automatically if you are actually using the respective fonts. This package ships with configuration files for a number of font families. Table 2 lists them all.

`\DeclareMicrotypeVariants` {*<list of suffixes>*}

`\DeclareMicrotypeVariants*` On its search for a configuration file, the package will also try to remove from the font name a suffix of one or more letters that denotes a ‘variant’ of the base font (cf. [Fontname](#)). This allows it to put settings for, e. g., the fonts `padx` (expert set), `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:

Table 2: Fonts with tailored protrusion settings

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

<sup>a</sup> Incomplete

<sup>b</sup> Aliases: Latin Modern (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor)

<sup>c</sup> Aliases: mathdesign/Charter (mdbch), MicroPress's chmath (chr)

<sup>d</sup> Settings inherited from italic shape

<sup>e</sup> Alias: mathdesign/URW Garamond (mdugm)

<sup>f</sup> Alias: ulgothic (ulg)

<sup>g</sup> By courtesy of Harald Harders ([h.harders@tu-bs.de](mailto:h.harders@tu-bs.de))

<sup>h</sup> Aliases: pxfonts (pxr), qfonts/QuasiPalatino, T<sub>E</sub>X Gyre Pagella (qp1), FPL Neu (fp9x, fp9j)

<sup>i</sup> Aliases: txfonts (txr), qfonts/QuasiTimes, T<sub>E</sub>X Gyre Termes (qtm)

<sup>j</sup> Alias: eulervm (zeur, zeus)

```
\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`.

<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.

## 6 Context-sensitive Setup

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

`\microtypecontext` {<context assignments>}

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

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

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

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

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

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

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

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

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

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

```
\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 has been loaded with the `babel` option (see section 3.5). Currently, `microtype` supports French and Turkish kerning and English spacing (aka. `\nonfrenchspacing`). For unknown languages, all contexts will be reset.

## 7 Letterspacing revisited

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

`\textls*` While the tracking feature, described in section 5.3, will apply to sets of fonts, you may also want to letterspace shorter pieces of text, regardless of the font in which they are typeset.<sup>10</sup> For such ad-hoc letterspacing, `microtype` introduces two commands that can be used in the same way as  $\text{\LaTeX}$ ’s text commands (independently of whether the tracking option is enabled): `\textls` expects the text in the mandatory argument, while `\lsstyle` will switch on letterspacing for all subsequent fonts until the end of the current group. The starred version of `\textls` does not add any extra space before or after the text, which may be useful, e. g., for section titles. By default, each character will be spaced out by 100/1000em (0.05em on each side); this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option, with decreasing significance in this order.

```
\lslig {<ligature>}
```

Since the commands `\textls` and `\lsstyle` will also evaluate the ‘no ligatures’ key for the respective font, you do not need to worry about protecting or breaking

<sup>10</sup> Letterspacing should be used cautiously. Renowned typographers have compared letterspacing lower-case text to stealing sheep (a capital offence in Britain in the 19th century). 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.

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, ‘`Uusſiđtsloſigfeit`’).

```
\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 (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.

## 8 Disabling Ligatures

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

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

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

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

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

The character that begins the ligature(s) is what matters. This command may only be used in the preamble, and only once.

## 9 Hints and Caveats

*Use settings that match your font.* Although the default settings should give reasonable results for most fonts, the particular font you happen to be using may have different character shapes that necessitate more or less protrusion or expansion. In particular, italic letter shapes may differ wildly in different fonts, hence I have decided against providing default protrusion settings for them.

The file `test-microtype.tex` might be of some help when adjusting the protrusion settings for a font.

*Don't use too large a value for expansion.* Font expansion is a feature that is supposed to enhance the typographic quality of your document by producing a more uniform greyness of the text block (and potentially reducing the number of necessary hyphenations). When expanding or shrinking a font too much, the effect will be turned into the opposite. Expanding the fonts by more than 2%, i. e., setting a stretch limit of more than 20, should be justified by a typographically trained eye. If you are so lucky as to be in the possession of multiple instances of a Multiple Master font, you may set expansion limits to up to 4%.

*Don't use font expansion for web documents (with older pdfTeX versions).* With pdfTeX versions older than 1.40, each expanded instance of the font will be embedded in the PDF file, hence the file size may increase by quite a large factor (depending on expansion limits and step). Therefore, courtesy and thriftiness of bandwidth command it not to enable font expansion when creating files to be distributed electronically. With pdfTeX 1.40, which uses a different technique of expansion, the file size increase can be neglected.

*Settings for Cyrillic/Greek/Thai etc. encodings are not yet included.* The default sets of fonts for which the micro-typographic features will be enabled (see table 1) only contain those encodings for which configurations exist. Therefore, if you are using any other encoding (e. g., T2A, LGR etc.), microtype will not apply to these fonts. You have to define and activate a new font set including the encoding(s) you are using (for details, see section 4). For protrusion, you would also have to create settings for the fonts in question (see section 5.1). It goes without saying that contributions for these encodings are more than welcome.

*Only employ kerning adjustment if it is customary in the language's typographic tradition.* In contrast to protrusion and expansion (and possibly adjustment of interword spacing), additional kerning does not unconditionally improve the micro-typographical quality of your document. You should only switch it on if you are writing a document in a language whose typographic tradition warrants such kerning. If you are, for example, writing an English text, your readers would probably be rather confused by additional spaces before the punctuation characters.

*You might want to disable protrusion in the Table of Contents.* In unfortunate situations, enabled protrusion might internally alter the line length in the TOC and



similar lists in such a way that an excess leader dot will fit in. The solution is to temporarily disable protrusion for the TOC:

```
\microtypesetup{protrusion=false}
\tableofcontents
\microtypesetup{protrusion=true}
```

*You might want to disable protrusion in verbatim environments.* As you know by now, `microtype` will by default activate character protrusion for all fonts contained in the font set ‘alltext’. This also includes the typewriter font. Although it does make sense to protrude the typewriter font if it appears in running text (like, for example, in this manual), this is probably not desirable inside the `verbatim` environment. However, `microtype` has no knowledge about the context that a font appears in but will solely decide by examining its attributes. Therefore, you have to take care of disabling protrusion in `verbatim` environments for yourself (that is, if you don’t want to disable protrusion for the typewriter font altogether, by choosing a different font set). While the `\microtypesetup` command has of course been designed for cases like this, you might find it tiring to repeat it every time if you are using the `verbatim` environment frequently. The following line, added to the document’s preamble, would serve the same purpose:<sup>11</sup>

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

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

- If you want to use 8-bit characters in the configuration, you have to load the `inputenc` package first. Unicode input is also supported (when loading `inputenc` with the `utf8` option, however, *not* with the `utf8x` option resp. the `ucs` package). If your document includes multiple input encodings, 8-bit characters in the settings will not work reliably unless you specify the `inputenc` key.

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

- ! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.

This error message will occur if you are trying to employ font expansion while creating DVI output. Remember, that *automatic* font expansion only works when running `pdf $\text{\TeX}$`  in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your  $\text{\TeX}$  system.

- ! pdf $\text{\TeX}$  error (font expansion): auto expansion is only possible with scalable fonts.

Automatic font expansion has been improved in `pdf $\text{\TeX}$`  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

<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.

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<sub>l</sub>atex: 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<sub>T</sub><sub>E</sub>X versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your <sub>T</sub><sub>E</sub>X system is not set up to embed (or 'download') the base PostScript fonts (e. g., Times, Helvetica, Courier). In most <sub>T</sub><sub>E</sub>X distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to true.
- Warning: pdf<sub>l</sub>atex (file ecrm1000+20): Font ecrm1000+20 at 1200 not found  
Furthermore, pdf<sub>T</sub><sub>E</sub>X versions older than 1.40 require Type 1 fonts for automatic font expansion. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. With older pdf<sub>T</sub><sub>E</sub>X versions, this is only possible if you manually create expanded instances of the fonts.
- ! Font T1/cmr/m/n/10=ecrm1000 at 10.0pt not loaded: Not enough room left.  
Memory parameter 'font\_mem\_size' too small.
- ! TeX capacity exceeded, sorry [maximum internal font number (font\_max)=2000].  
Memory parameter 'font\_max' too small.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf\_mem\_size)=65536].  
Memory parameter 'pdf\_mem\_size' too small (pdf<sub>T</sub><sub>E</sub>X versions older than 1.30).  
When applying micro-typographic enhancement to a large document with a lot of fonts, pdf<sub>T</sub><sub>E</sub>X may be running out of some kind of memory. It can be increased by setting the respective parameter to a larger value. For web2c-based systems, change the settings in `texmf.cnf`, for MiK<sub>T</sub><sub>E</sub>X, in the file `miktex.ini` (2.4 or older) resp. `pdflatex.ini` (2.5 or newer).
- pdf<sub>T</sub><sub>E</sub>X warning (font expansion): font should be expanded before its first use  
This warning will occur with pdf<sub>T</sub><sub>E</sub>X versions older than 1.40.4, if tracking *and* expansion is applied to a font. It is harmless and can be ignored.

## 10 Contributions

I would be glad to include configuration files for more fonts. Preparing such configurations is quite a time-consuming task and requires a lot of patience. To alleviate this process, this package also includes a test file that can be used to check at least the protrusion settings (`test-microtype.tex`).

If you have created a configuration file for another font, or if you have any suggestions for enhancements in the default configuration files, I would gratefully accept them: [w.m.l@gmx.net](mailto:w.m.l@gmx.net).

## 11 Acknowledgments

This package would be pointless if *Hàn Thế Thành* hadn't created the pdf<sub>T</sub><sub>E</sub>X programme in the first place, which introduced the micro-typographic extensions

and made them available to the  $\text{\TeX}$  world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004]. I also thank him and the rest of the pdf $\text{\TeX}$  team for refuting the idea that  $\text{\TeX}$  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 and Steven E. Harris.

## 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, 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 pdfcpot.sty package*, 10 June 2005. (Available from CTAN at </macros/latex/contrib/pdfcpot/>)

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

## 13 Short History

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

### 2.2 (2007/07/14)

- Improvements to tracking/letterspacing: retain kerning (with pdfTeX 1.40.4); automatically adjust protrusion settings
- New key ‘no ligatures’ for `\SetTracking` to disable selected or all ligatures (with pdfTeX 1.40.4) (see section 5.3)
- New keys ‘spacing’ and ‘outer spacing’ for `\SetTracking` to customise interword spacing (see section 5.3)
- Possibility to expand a font with different parameters (with pdfTeX 1.40.4) (see section 5.2)
- New optional argument for `\DisableLigatures` to disable selected ligatures only (see section 8)
- New command `\DeclareMicrotypeVariants` to specify variant suffixes (see section 5.7)
- New command `\textmicrotypecontext` as a wrapper for `\microtypecontext` (see 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 (see 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’; see sections 5.3, 5.4 and 5.5)
- New commands `\textls` and `\lstyle` for letterspacing, new option ‘letterspace’ (see sections 3.4 and 7)
- New option ‘babel’ for automatic micro-typographic adjustment to the selected language (see sections 3.5 and 6)
- New font sets: ‘smallcaps’, ‘footnotesize’, ‘scriptsize’ (see section 4 and table 1)
- New package ‘letterspace’ providing the commands for robust and hyphenatable letterspacing

### 1.9e (2006/07/28)

- New key ‘inputenc’ to specify the lists’ input encodings (see 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 (see 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; see section 8)
- New command `\microtypecontext` to change the configuration context; new key ‘context’ for the configuration commands (see section 6)
- New key ‘font’ to add single fonts to the font sets (see 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)
- ‘DVOutput’ option will work with `TeXLive 2004`

## 1.8 (2005/06/23)

- If font substitution has occurred, the settings for the substitute will be used instead of those for the selected font
- New command `\DeclareMicrotypeSetDefault` to declare the default font sets (see section 4)
- New option ‘config’ to load a different configuration file (see section 3.5)
- New option ‘unit’ to measure protrusion factors relative to a dimension instead of the character width (see section 5.1)
- Renamed commands from `\..MicroType..` to `\..Microtype..`
- Protrusion settings for AMS math fonts
- Protrusion settings for Times in LY1 encoding completed
- The ‘allmath’ font set also includes U encoding
- 8-bit characters in the configuration finally work as advertised, even if made active by the `csquotes` package
- When using the `ledmac` package, character protrusion will work for the first time ever (requires `pdfTeX` version 1.30 or later)

## 1.7 (2005/03/23)

- Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings (see sections 4 and 5)
- Always take font size into account when trying to find protrusion resp. expansion settings for a given font (see section 5)
- New command `\LoadMicrotypeFile` to load a font configuration file manually (see section 5.7)

- Hook `\Microtype@Hook` for font package authors (see section 14.4.3)
- New option `'verbose=errors'` to turn all warnings into errors
- Disable expansion inside `\showhyphens`
- Warning when running in draft mode

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

- New option `'factor'` to influence protrusion resp. expansion of all characters of a font or font set (see sections 3.2 and 5)
- When pdf $\TeX$  is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled (see section 3.1)
- Protrusion settings of digits improved
- Use e- $\TeX$  extensions, if available

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

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

#### 1.4b (2004/11/26)

- `\UseMicrotypeSet` requires the set to be declared (see section 4)

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

- Set up fonts independently from  $\LaTeX$  font loading (therefore, no risk of overlooking fonts anymore, and the package may be loaded at any time)
- `\microtypesetup` now sets the correct level of protrusion (see chapter 3.6)
- New option: `'final'`

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

- New font sets: `'allmath'` and `'basicmath'` (see section 4 and table 1)
- Protrusion settings for Computer Modern Roman math symbols
- Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond
- If an alias font name is specified, it will be used as an alternative, not as a replacement (see section 5.7)

#### 1.1 (2004/09/21)

- Protrusion settings for Adobe Minion, contributed by Harald Harders
- New command: `\DeclareCharacterInheritance` (see section 5.6)
- Characters may also be specified as octal or hexadecimal numbers (see section 5)
- Configuration file names in lowercase (see section 5.7)

#### 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).

debug: Code for additional output in the log file.

Used for – surprise! – debugging purposes.

lua: Code for Lua<sub>T</sub><sub>E</sub>X (by default disabled)

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

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.

- 1 `<*package|letterspace>`
- 2 `<*package>`

## 14.1 Preliminaries

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

```

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

```

These commands also have a starred version.

```

28 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
29 \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.

```

30 \@onlypreamble\DeclareMicrotypeSet
31 \@onlypreamble\UseMicrotypeSet
32 \@onlypreamble\DeclareMicrotypeSetDefault
33 \@onlypreamble\DisableLigatures
34 \@onlypreamble\DeclareMicrotypeVariants
35 \@onlypreamble\DeclareMicrotypeBabelHook

```

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

```

36 \def\MT@old@cmd#1#2{%
37   \newcommand*#1{\MT@warning{%
38     \string#1 is deprecated. Please use\MessageBreak
39     \string#2 instead}%
40     \let #1#2#2}}
41 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
42 \MT@old@cmd\DeclareMicroTypeSet \DeclareMicrotypeSet
43 \MT@old@cmd\UseMicroTypeSet \UseMicrotypeSet
44 \MT@old@cmd\LoadMicroTypeFile \LoadMicrotypeFile
45 </package>

```

\MT@MT      This is us.

```

46 \def\MT@MT
47 <package> {microtype}

```



```

48 <letterspace> {letterspace}

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

```

### 14.1.1 Debugging

\tracingmicrotype Cases for \tracingmicrotype:

```

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

```

```

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

```

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

Cases for \tracingmicrotypeinpdf:

```

1: show new fonts
2: + show known fonts

```

```
71 \newcount\tracingmicrotypeinpdf
```

Let's see how it works ...

```
\tracingmicrotypeinpdf=2
```

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

```

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

```

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

77 `\newif\iftracingmicrotypeinpdfall`

`\MT@show@pdfannot` A red caret is shown for fonts which are actually set up, a green one marks fonts that have already been seen. The `/Caret` annotation requires a viewer for PDF version 1.5 (you could use `/Text` if you're using an older PDF viewer).

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

### 14.1.2 Requirements

`\MT@pdftex@no` pdf<sub>T</sub><sub>E</sub>X's features for which we provide an interface here haven't always been available, and some specifics have changed over time. Therefore, we have to test which pdf<sub>T</sub><sub>E</sub>X we're using, if any. `\MT@pdftex@no` will be used throughout the package to respectively do the right thing.

Currently, we have to distinguish seven cases for pdf<sub>T</sub><sub>E</sub>X:

- 0: not running pdf<sub>T</sub><sub>E</sub>X
- 1: pdf<sub>T</sub><sub>E</sub>X (< 0.14f)
- 2: + micro-typographic extensions (0.14f, 0.14g)
- 3: + protrusion relative to 1em ( $\geq$  0.14h)
- 4: + automatic font expansion; default `\efcode = 1000` ( $\geq$  1.20)
- 5: + `\(left,right)marginkern`; `\pdfnoligatures`; `\pdfstrcmp`; `\pdfescapestring` ( $\geq$  1.30)
- 6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdftracingfonts`; `\pdfmatch`<sup>12</sup>; always e-<sub>T</sub><sub>E</sub>X ( $\geq$  1.40)
- 7: + `\letterspacefont` doesn't disable ligatures and kerns; `\pdfcopyfont` ( $\geq$  1.40.4)

96 </package>

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

<sup>12</sup> This command was actually introduced in 1.30, but was still buggy then.

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

```

98 \ifx\normalpdftexversion\@undefined \else
99   \let\pdftexversion\normalpdftexversion
100   \let\pdftexrevision\normalpdftexrevision
101   \let\pdfoutput\normalpdfoutput
102 \fi

```

Old packages might have let \pdftexversion to \relax.

```

103 \ifx\pdftexversion\@undefined \else
104   \ifx\pdftexversion\relax \else
105     <debug>\MT@info@n1{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
106     \def\MT@pdftex@no{7}
107     <*package>
108       \ifnum\pdftexversion = 140
109         \ifnum\pdftexrevision < 4
110           \def\MT@pdftex@no{6}
111         \fi
112       \else
113     </package>
114       \ifnum\pdftexversion < 140
115         \def\MT@pdftex@no{5}
116     <*package>
117       \ifnum\pdftexversion < 130
118         \def\MT@pdftex@no{4}
119       \ifnum\pdftexversion < 120
120         \def\MT@pdftex@no{3}
121       \ifnum\pdftexversion = 14
122         \ifnum\expandafter`\pdftexrevision < `h
123           \def\MT@pdftex@no{2}
124         \ifnum\expandafter`\pdftexrevision < `f
125           \def\MT@pdftex@no{1}
126         \fi
127       \fi
128     \else
129       \ifnum\pdftexversion < 14
130         \def\MT@pdftex@no{1}
131       \fi
132     \fi
133   \fi
134 \fi
135 \fi
136 </package>
137 \fi
138 \fi
139 \fi
140 <debug>\MT@info@n1{0}{pdftex no.: \MT@pdftex@no}

```

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

```

141 \def\MT@clear@options{%
142   \AtEndOfPackage{\let\@unprocessedoptions\relax}%
143   \let\CurrentOption\@empty
144 }

145 \ifnum\MT@pdftex@no <
146 <package> 2
147 <letterspace> 6
148   \MT@clear@options
149   \MT@warning@n1{You
150     \ifcase\MT@pdftex@no

```

```

151     don't seem to be using pdftex.\MessageBreak
152     ~\MT@MT' only works with pdftex.\MessageBreak
153     Try running 'pdflatex' instead of 'latex'%
154     \else
155         are using a pdftex version older than
156         <package>      0.14f%
157         <letterspace>  1.40%
158         .\MessageBreak
159         ~\MT@MT' does not work with this version.\MessageBreak
160         Please install a newer version of pdftex%
161     \fi
162 }
163 \endinput\fi
164 <*package>

```

Still there? Then we can begin:

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

```

165 \def\MT@catcodes{%
166   \catcode~\^7 %
167   \@makeother\~%
168   \@makeother\=%
169   \@makeother\*%
170   \@makeother\+%
171   \@makeother\,%
172   \@makeother\/%
173   \@makeother\`%
174   \@makeother\'%
175   \@makeother\"%
176   \@makeother\!%
177 }

```

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

```

178 \def\MT@restore@catcodes#1{%
179   \ifx\relax#1\else
180     \noexpand\catcode~\noexpand#1\the\catcode~#1\relax
181     \expandafter\MT@restore@catcodes
182   \fi
183 }
184 \edef\MT@restore@catcodes{\MT@restore@catcodes\^-\=\*\+\,\/\`\'\"!\relax}
185 \MT@catcodes
186 \AtEndOfPackage{\MT@restore@catcodes}

```

We need the keyval package, including the 'new' \KV@@sp@def implementation.

```

187 </package>
188 \RequirePackage{keyval}[1997/11/10]
189 <*package>

```

\MT@toks We need a token register.

```
190 \newtoks\MT@toks
```

\ifMT@if@ A scratch if.

```
191 \newif\ifMT@if@
```

### 14.1.3 Declarations

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

```

\ifMT@expansion 192 \newif\ifMT@protrusion
\ifMT@auto
\ifMT@selected
\ifMT@noligatures
\ifMT@draft
\ifMT@spacing
\ifMT@kerning
\ifMT@tracking
\ifMT@babel

```

```

193 \newif\ifMT@expansion
194 \newif\ifMT@auto
195 \newif\ifMT@selected
196 \newif\ifMT@noligatures
197 \newif\ifMT@draft
198 \newif\ifMT@spacing
199 \newif\ifMT@kerning
200 \newif\ifMT@tracking
201 \newif\ifMT@babel

\MT@pr@level    ... and numbers.
\MT@ex@level    202 \let\MT@pr@level\tw@
\MT@pr@factor   203 \let\MT@ex@level\tw@
\MT@ex@factor   204 \let\MT@pr@factor\@m
\MT@ex@factor   205 \let\MT@ex@factor\@m
\MT@sp@factor   206 \let\MT@sp@factor\@m
\MT@kn@factor   207 \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     208 \let\MT@pr@unit\@empty
\MT@st@stretch  209 \let\MT@sp@unit\m@ne
\MT@shrink      210 \def\MT@kn@unit{1em}
\MT@step        Expansion settings.
                211 \let\MT@stretch\m@ne
                212 \let\MT@shrink \m@ne
                213 \let\MT@step \m@ne

\MT@pr@min      Minimum and maximum values allowed by pdfTEX.
\MT@pr@max      214 \def\MT@pr@min{-\@m}
\MT@ex@min      215 \let\MT@pr@max\@m
\MT@ex@min      216 \let\MT@ex@min\z@
\MT@ex@max      217 \let\MT@ex@max\@m
\MT@sp@min      218 \def\MT@sp@min{-\@m}
\MT@sp@max      219 \let\MT@sp@max\@m
\MT@kn@min      220 \def\MT@kn@min{-\@m}
\MT@kn@min      221 \let\MT@kn@max\@m
\MT@kn@max      222 </package>
\MT@tr@min      223 \def\MT@tr@min{-\@m}
\MT@tr@min      224 \let\MT@tr@max\@m
\MT@tr@max      225 <*package>

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

\MT@stretch@default    Default values for expansion.
\MT@shrink@default     227 \def\MT@stretch@default{20 }
\MT@step@default       228 \def\MT@shrink@default{20 }
                      229 \def\MT@step@default{4 }

\MT@letterspace        Default value for letterspacing (in thousandths of 1em).
\MT@letterspace@default 230 </package>
                      231 \let\MT@letterspace\m@ne
                      232 \def\MT@letterspace@default{100}
                      233 <*package>

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

```

### 14.1.4 Auxiliary Macros

`\MT@maybe@etex` For definitions that depend on e- $\TeX$  features.

```

\MT@maybe@etex@ 235 \def\MT@maybe@etex{%
236   \ifcase 0%
237     \ifx\TeXversion\undefined 1\else
238       \ifx\TeXversion\relax 1\else
239         \ifcase\TeXversion 1\fi
240       \fi
241     \fi
242     \catcode`Q=14 \catcode`X=9
243   \else
244     \catcode`Q=9 \catcode`X=14
245   \fi
246   \MT@maybe@etex@
247 }
248 \def\MT@maybe@etex@#1{#1%
249   \catcode`Q=11 \catcode`X=11
250 }
251 <debug>\MT@maybe@etex{\MT@dinfol{0}{this is
252 <debug>Q not
253 <debug> etex}}
```

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

```

254 \def\MT@requires@pdftex#1{%
255   \ifnum\MT@pdftex@no<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
256 }
257 <debug>\MT@requires@pdftex6{\pdftracingfonts=1}\relax
```

`\MT@requires@luatex` For definitions that depend on Lua $\TeX$ .

```

258 <*lua>
259 \def\MT@requires@luatex{%
260   \ifx\directlua\undefined\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
261 }
262 <debug>\MT@dinfol{0}{this is \MT@requires@luatex{}{not }luatex}
```

`\MT@lua` Is zero the correct number for the lua state?

```

263 \def\MT@lua{\directlua{z@}}
264 </lua>
265 </package>
```

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

```

266 <*!lua>
267 \ifx\directlua\undefined\else
268 <!letterspace> \MT@error
269 <letterspace> \MT@warning@n1
270   {\MT@MT'
271 <!letterspace> only works with luatex if you generate\MessageBreak
272 <!letterspace> the package with the `lua' option%
273 <letterspace> doesn't currently work with luatex.\MessageBreak Bye bye%
274 }
275 <!letterspace> {}
276 <letterspace> \MT@clear@options\expandafter\endinput
277 \fi
278 </!lua>
```

`\MT@glet` The forgotten primitive.

```

279 \def\MT@glet{\global\let}
280 <*package>
```

```

\MT@def@n      This is \@namedef and global.
\MT@gdef@n 281 \def\MT@def@n#1{\expandafter\def\csname #1\endcsname}
282 \def\MT@gdef@n#1{\expandafter\gdef\csname #1\endcsname}

\MT@edef@n      Its expanding versions.
\MT@xdef@n 283 </package>
284 \def\MT@edef@n#1{\expandafter\edef\csname #1\endcsname}
285 <*package>
286 \def\MT@xdef@n#1{\expandafter\xdef\csname #1\endcsname}

\MT@let@nc      \let a \csname sequence to a command.
287 \def\MT@let@nc#1{\expandafter\let\csname #1\endcsname}

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

\MT@let@nn      \let a \csname sequence to a \csname sequence.
289 \def\MT@let@nn#1{\expandafter\MT@let@cn\csname #1\endcsname}

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

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

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

\MT@exp@two@n      Expand the next two tokens after <#1> once and enclose them in braces.
295 \def\MT@exp@two@n#1#2#3{%
296   \expandafter\expandafter\expandafter
297   #1\expandafter\expandafter\expandafter
298   {\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 299 \MT@maybe@etex{
300   \def\MT@ifdefined@c@T#1{%
301     X   \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
302     Q   \ifx#1\undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
303   }
304 </package>
305 <letterspace>\catcode`\X=9
306   \def\MT@ifdefined@c@TF#1{%
307     X   \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
308     <package>Q   \ifx#1\undefined
309     <package>Q     \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
310   }
311   \def\MT@ifdefined@n@T#1{%
312     X   \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
313     <package>Q   \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
314     <package>Q     \expandafter\@gobble\else\expandafter\@firstofone\fi
315   }
316 <letterspace>\catcode`\X=11
317 <*package>
318   \def\MT@ifdefined@n@TF#1{%

```

```

319 X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
320 Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
321 Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
322 }
323 }

```

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

```

324 \MT@maybe@etex{
325 \def\MT@detokenize@n#1{%
326 X \detokenize\expandafter{%
327 \string#1%
328 X }%
329 }
330 \def\MT@detokenize@c#1{%
331 X \detokenize\expandafter\expandafter\expandafter{\expandafter\string#1}%
332 Q \MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty
333 }
334 }

```

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

```

335 </package>
336 \begingroup
337 \catcode`\%=12
338 \catcode`\&=14
339 \gdef\MT@ifempty#1{%
340 \if %#1%&
341 \expandafter\@firstoftwo
342 \else
343 \expandafter\@secondoftwo
344 \fi
345 }
346 \endgroup
347 <*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).

```

348 \MT@requires@pdftex6{
349 <*lua>
350 \MT@requires@luatex{
351 \def\MT@ifint#1{%
352 \csname
353 \MT@lua{if string.find("\luaescapestring{#1}", "^~*[0-9]+ *$")
354 then tex.write("@firstoftwo")
355 else tex.write("@secondoftwo")
356 end}%
357 \endcsname
358 }
359 }{
360 </lua>
361 </package>
362 \def\MT@ifint#1{%
363 \ifcase\pdfmatch{^~*[0-9]+ *$}{#1}\relax
364 \expandafter\@secondoftwo
365 \else
366 \expandafter\@firstoftwo
367 \fi
368 }
369 <*package>

```



```

370 <lua> }
371 }{
372   \def\MT@ifint#1{%
373     \if!\ifnum9<1#1!\else?\fi
374     \expandafter\@firstoftwo
375     \else
376     \expandafter\@secondoftwo
377     \fi
378   }
379 }

\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.)

380 \MT@requires@pdftex6{
381 <*lua>
382   \MT@requires@luatex{
383     \def\MT@ifdimen#1{%
384       \csname % fix expression!
385         \MT@lua{if (string.find("\luaescapestring{#1}", "^-[0-9]+(\@percentchar a*) *$") or
386           string.find("\luaescapestring{#1}", "^-[0-9]*[.][0-9]+(\@percentchar a*) *$"))
387           then tex.write("@firstoftwo")
388           else tex.write("@secondoftwo")
389         end}%
390       \endcsname
391     }
392   }{
393 </lua>
394     \def\MT@ifdimen#1{%
395       \ifcase\pdfmatch{^[0-9]+([.][0-9]+)?|([.][0-9]+)%
396         (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
397       \expandafter\@secondoftwo
398       \else
399       \expandafter\@firstoftwo
400       \fi
401     }
402 <lua> }
403 }{
404   \def\MT@ifdimen#1{%
405     \setbox\z@=\hbox{%
406       \MT@count=1#1\relax
407       \ifnum\MT@count=\@ne
408         \aftergroup\@secondoftwo
409       \else
410         \aftergroup\@firstoftwo
411       \fi
412     }%
413   }
414 }

\MT@ifdim      Test floating point numbers.

415 \def\MT@ifdim#1#2#3{%
416   \ifdim #1\p@ #2 #3\p@
417     \expandafter\@firstoftwo
418   \else
419     \expandafter\@secondoftwo
420   \fi
421 }

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

422 \MT@requires@pdftex5{
423 <*lua>
424   \MT@requires@luatex{

```

```

425 \def\MT@ifstreq#1#2{%
426 \csname
427 \MT@lua{if "\luaescapestring{#1}" == "\luaescapestring{#2}"
428 then tex.write("@firstoftwo")
429 else tex.write("@secondoftwo")
430 end}%
431 \endcsname
432 }
433 }{
434 /lua
435 \def\MT@ifstreq#1#2{%
436 \ifcase\pdfstrcmp{#1}{#2}\relax
437 \expandafter\@firstoftwo
438 \else
439 \expandafter\@secondoftwo
440 \fi
441 }
442 <lua> }
443 }{
444 \def\MT@ifstreq#1#2{%
445 \edef\MT@res@a{#1}%
446 \edef\MT@res@b{#2}%
447 \ifx\MT@res@a\MT@res@b
448 \expandafter\@firstoftwo
449 \else
450 \expandafter\@secondoftwo
451 \fi
452 }
453 }

```

`\MT@xadd` Add item to a list.

```

454 \def\MT@xadd#1#2{%
455 \ifx#1\relax
456 \xdef#1{#2}%
457 \else
458 \xdef#1{#1#2}%
459 \fi
460 }

```

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

```

461 \def\MT@xaddb#1#2{%
462 \ifx#1\relax
463 \xdef#1{#2}%
464 \else
465 \xdef#1{#2#1}%
466 \fi
467 }

```

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

`\MT@map@clist@` *</package>*

```

\MT@clist@function 468 \def\MT@map@clist@n#1#2{%
\MT@clist@break 469 \ifx\@empty#1\else
470 \def\MT@clist@function##1{#2}%
471 \MT@map@clist@#1,\@nil,\@nnil
472 \fi
473 }
474 }
475 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
476 \def\MT@map@clist@#1,%
477 \ifx\@nil#1%
478 \expandafter\MT@clist@break
479 \fi

```

```

480 \MT@clist@function{#1}%
481 \MT@map@clist@
482 }
483 \let\MT@clist@function\@gobble
484 \def\MT@clist@break#1\@nnil{}
485 *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@ 486 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break 487 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
488 \def\MT@map@tlist@#1#2{%
489   \ifx\@nnil#2\else
490     #1{#2}%
491     \expandafter\MT@map@tlist@
492     \expandafter#1%
493   \fi
494 }
495 \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 496 \newif\ifMT@inlist@
497 \def\MT@in@clist#1#2{%
498   \def\x##1,#1,##2##3\@nnil{%
499     \ifx##2\@empty
500       \MT@inlist@false
501     \else
502       \MT@inlist@true
503     \fi
504   }%
505   \expandafter\x\expandafter,#2,#1,\@empty\@nnil
506 }

```

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

```

507 \def\MT@rem@from@clist#1#2{%
508   \def\x##1,#1,##2\x{##1,##2\y}%
509   \def\y##1,\y##2\y{\ifx,##1\@empty\else##1\fi}%
510   \xdef#2{\MT@exp@two@c\y\x\expandafter,#2,\y,#1,\x}%
511 }

```

\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.

```

512 \def\MT@in@tlist#1#2{%
513   \MT@inlist@false
514   \def\x{#1}%
515   \MT@map@tlist@c#2\MT@in@tlist@
516 }
517 \def\MT@in@tlist@#1{%
518   \edef\y{#1}%
519   \ifx\x\y
520     \MT@inlist@true
521     \expandafter\MT@tlist@break
522   \fi
523 }

```

\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@@ 524 \def\MT@in@rlist#1{%
\MT@size@name 525   \MT@inlist@false

```

```

526 \MT@map@tlist@c#1\MT@in@rlist@
527 }
528 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
529 \def\MT@in@rlist@@#1#2#3{%
530 \MT@ifdim{#2}=\m@ne{%
531 \MT@ifdim{#1}=\MT@size
532 \MT@inlist@true
533 \relax
534 }%
535 \MT@ifdim\MT@size<{#1}\relax%
536 \MT@ifdim\MT@size<{#2}%
537 \MT@inlist@true
538 \relax
539 }%
540 }%
541 \ifMT@inlist@
542 \def\MT@size@name{#3}%
543 \expandafter\MT@tlist@break
544 \fi
545 }

\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 546 \def\MT@loop#1\MT@repeat{%
547 \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
548 \MT@iterate \let\MT@iterate\relax
549 }
550 \let\MT@repeat\fi

\MT@while@num Execute <#3> from <#1> up to (excluding) <#2>.
551 \def\MT@while@num#1#2#3{%
552 \@tempcnta#1\relax
553 \MT@loop #3%
554 \advance\@tempcnta \@ne
555 \ifnum\@tempcnta < #2\MT@repeat
556 }

\MT@do@font Execute <#1> 256 times.
557 \def\MT@do@font{\MT@while@num\z@\cc@lvi}

\MT@count Increment macro <#1> by one. Saves using up too many counters. The e-TEX way is
\MT@increment slightly faster.
558 \newcount\MT@count
559 \MT@maybe@etex{
560 \def\MT@increment#1{%
561 X \edef#1{\number\numexpr #1 + 1\relax}%
562 Q \MT@count=#1\relax
563 Q \advance\MT@count \@ne
564 Q \edef#1{\number\MT@count}%
565 }

\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.
566 \def\MT@scale#1#2#3{%
567 Q \multiply #1 #2\relax
568 \ifnum #3 = \z@
569 X #1=\numexpr #1 * #2\relax
570 \else
571 X #1=\numexpr #1 * #2 / #3\relax

```

```

572 Q      \divide #1 #3\relax
573      \fi
574   }
575 }

```

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

```

\MT@abbr@pr@c 576 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 577 \def\MT@abbr@ex{expansion}
\MT@abbr@pr@c 578 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@pr@inh 579 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh 580 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@ex@inh 581 \def\MT@abbr@ex@inh{expansion inheritance}
\MT@abbr@n 582 \def\MT@abbr@n{noligatures}
\MT@abbr@sp 583 \def\MT@abbr@sp{spacing}
\MT@abbr@sp@c 584 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@sp@inh 585 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@sp@inh 586 \def\MT@abbr@kn{kerling}
\MT@abbr@kn 587 \def\MT@abbr@kn@c{kerling codes}
\MT@abbr@kn@c 588 \def\MT@abbr@kn@inh{kerling inheritance}
\MT@abbr@kn@c 589 \def\MT@abbr@tr{tracking}
\MT@abbr@kn@inh 590 \def\MT@abbr@tr@c{tracking amount}
\MT@abbr@tr
\MT@rbb@protrusion
\MT@rbb@tr@c
\MT@rbb@expansion 591 \def\MT@rbb@protrusion{pr}
\MT@rbb@spacing 592 \def\MT@rbb@expansion{ex}
\MT@rbb@spacing 593 \def\MT@rbb@spacing{sp}
\MT@rbb@kerling 594 \def\MT@rbb@kerling{kn}
\MT@rbb@tracking 595 \def\MT@rbb@tracking{tr}

```

These we also need the other way round.

```

\MT@features
\MT@features@long 596 \def\MT@features{pr,ex,sp,kn,tr}
597 \def\MT@features@long{protrusion,expansion,spacing,kerling,tracking}

```

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

```

598 \def\MT@is@feature#1{%
599   \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
600   \ifMT@inlist@
601     \expandafter\@firstofone
602   \else
603     \MT@error{`\@tempa' is not an available micro-typographic\MessageBreak
604       feature. Ignoring #1}{Available features are: `~\MT@features@long'.}%
605     \expandafter\@gobble
606   \fi
607 }

```

### 14.1.5 Compatibility

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

- `\pickup@font`
- `\do@subst@correction`
- `\add@accent`
- `\showhyphens`

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

```
608 \ifloaded{tex}{wordcount}{%
609   \MT@warning@nl{Detected the `wordcount' utility.\MessageBreak
610     Disabling `MT@MT', since it wouldn't work}%
611   \MT@clear@options\endinput}\relax
```

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

```
612 </package>
613 \let\MT@setup@\@empty
```

`\MT@addto@setup` We use our private hook to have better control over the timing.

```
614 \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).

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

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

```
616 \def\MT@with@package@T#1{\ifpackageloaded{#1}\@firstofone\@gobble}
617 <*package>
```

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

```
618 \def\MT@with@babel@and@T#1{%
619   \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
620     \@expandtwoargs\MT@in@clist{#1}
621     {\cename opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
622     \ifMT@inlist\@expandafter\@gobble\fi
623   }\@gobble
624 }
```

Don't load letterspace.

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

`\MT@ledmac@setup` The `ledmac` package first saves each paragraph in a box, from which it then splits  
`\MT@led@unhbox@line` off the lines one by one. This will destroy character protrusion. (There aren't any  
`\MT@led@kern` problems with the `lineno` package, since it takes a different approach.) — ... —  
 After much to and fro, the situation has finally settled and there is a fix. Beginning  
 with pdf<sub>T</sub>E<sub>X</sub> version 1.21b together with `ledpatch.sty` as of 2005/06/02 (v0.4),  
 character protrusion will work at last.

Peter Wilson was so kind to provide the `\l@dunhbox@line` hook in `ledmac` to allow for protrusion. `\leftmarginkern` and `\rightmarginkern` are new primitives of pdf<sub>T</sub>E<sub>X</sub> 1.21b (aka. 1.30.0).

```
626 \MT@requires@pdftex5{
627   \def\MT@ledmac@setup{%
628     \ifMT@protrusion
629       \MT@ifdefined@c@TF\l@dunhbox@line{%
630         \MT@info@nl{Patching ledmac to enable character protrusion}%
631         \newdimen\MT@led@kern
632         \let\MT@led@unhbox@line\l@dunhbox@line
633         \renewcommand*\l@dunhbox@line}[1]{%
634           \ifhbox##1%
635             \MT@led@kern=\rightmarginkern##1%
636           \kern\leftmarginkern##1%
```

```

637      \MT@led@unhbox@line##1%
638      \kern\MT@led@kern
639      \fi
640    }%
641  }{%
642    \MT@warning@nl{%
643      Character protrusion in paragraphs with line\MessageBreak
644      numbering will only work if you update ledmac}%
645  }%
646  \fi
647 }
648 }{
649 \def\MT@ledmac@setup{%
650   \ifMT@protrusion
651     \MT@warning@nl{%
652       The pdftex version you are using does not allow\MessageBreak
653       character protrusion in paragraphs with line\MessageBreak
654       numbering by the 'ledmac' package.\MessageBreak
655       Upgrade pdftex to version 1.30 or later}%
656   \fi
657 }
658 }

```

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

```

659 \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.)

```

660 \def\MT@setupfont@hook{%

```

Spanish and Galician babel modify \%, storing the original meaning in \percentsign.

```

661   \MT@with@babel@and@T{spanish} {\MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
662   \MT@with@babel@and@T{galician}{\MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%

```

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

```

663   \MT@with@package@T{csquotes}
664   {\@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax}%

```

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

```

665   \@ifpackageloaded{hyperref}{\MT@restore@p@h
666     {\MT@with@package@T{tex4ht}\MT@restore@p@h}%
667 }

```

Check again at the end of the preamble.

```

668 </package>
669 \MT@addto@setup{%
670 <*package>

```

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

```

671   \MT@with@package@T{pdfcpot}{%

```

```

672 \MT@error{Detected the `pdfcprot' package!\MessageBreak
673         ~\MT@MT' and `pdfcprot' may not be used together}{%
674 The `pdfcprot' package provides an interface to character protrusion.\MessageBreak
675 So does the ~\MT@MT' package. Using both packages at the same\MessageBreak
676 time will almost certainly lead to undesired results. Have your choice!}%
677 }
678 \MT@with@package@T{ledmac}\MT@ledmac@setup

```

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

```

679 \let\MT@setupfont@hook\@empty
680 \MT@with@babel@and@T{spanish}{%
681   \g@addto@macro\MT@setupfont@hook{%
682     \MT@ifdefined@c@T\percentsign{\let%\percentsign}}%
683   }
684 \MT@with@babel@and@T{galician}{%
685   \g@addto@macro\MT@setupfont@hook{%
686     \MT@ifdefined@c@T\percentsign{\let%\percentsign}}%
687   }
688 \MT@with@package@T{csquotes}{%
689   \ifpackageafter{csquotes}{2005/05/11}{%
690     \g@addto@macro\MT@setupfont@hook\@disablequotes
691   }{%
692     \MT@warning@n{%
693       Should you receive warnings about unknown slot\MessageBreak
694       numbers, try upgrading the `csquotes' package}%
695   }%
696 }
697 \ifpackageafterloaded
698 </package>
699 <letterspace> \MT@with@package@T
700   {hyperref}{%
701 </package>   \g@addto@macro\MT@setupfont@hook\MT@restore@p@h

```

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

```

702 \pdfstringdefDisableCommands{%
703 <package>   \let\pickup@font\MT@orig@pickupfont
704   \def\lststyle{\pdfstringdefWarn\lststyle}%
705   \def\textls#1#\pdfstringdefWarn\textls}%
706 }%
707 }
708 <*package>
709   {\MT@with@package@T{tex4ht}
710    {\g@addto@macro\MT@setupfont@hook\MT@restore@p@h}}%

```

The listings package makes numbers and letters active,

```

711 \MT@with@package@T{listings}{%
712   \g@addto@macro\MT@normal@catcodes{%
713     \MT@while@num{"30}{\catcode\@tempcnta 12\relax}%
714     \MT@while@num{"41}{\catcode\@tempcnta 11\relax}%
715     \MT@while@num{"61}{\catcode\@tempcnta 11\relax}%
716   }%

```

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

```

717   \g@addto@macro\MT@setupfont@hook{%
718     \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.

```

719   \let\lst@ProcessLetter\@empty
720 }%

```



721 }

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.

```
722 %
723 </package>
724 \MT@with@package@T{soul}{%
725   \soulregister\lsstyle 0%
726   \soulregister\textls 1%
727 }
728 <*package>
```

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

```
729 \MT@with@package@T{pinyin}{%
730   \let\MT@orig@py@macron\py@macron
731   \ifpackageafter{pinyin}{2006/10/17}{% 4.7.0
732     \def\py@macron#1#2{%
733       \let\pickup@font\MT@orig@pickupfont
734       \MT@orig@py@macron{#1}{#2}%
735       \let\pickup@font\MT@pickupfont}%
736     }{%
737       \def\py@macron#1{%
738         \let\pickup@font\MT@orig@pickupfont
739         \MT@orig@py@macron{#1}%
740         \let\pickup@font\MT@pickupfont}%
741       }%
742     }
743 </package>
744 }
745 <*package>
```

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

```
746 \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`).

```
747 \def\MT@setupfont{%
```

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

```
748 \MT@setupfont@hook
749 }
```

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

```
750 \MT@requires@pdftex7
751 {\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!

```
752 \g@addto@macro\MT@setupfont{%
753   \MT@exp@two@{c}\MT@split@name\string\MT@font/\@nil
```

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

```
754 \MT@exp@one@{n}\MT@find@file\MT@family
755 \ifx\MT@familyalias\empty \else
```

```
756 \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.)

```
757 % \ifx\fontencoding\cfencoding\else\@enc@update\fi
758 }
```

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

```
759 \MT@requires@pdftex6
760 {\g@addto@macro\MT@setupfont\MT@tracking}\relax
761 \g@addto@macro\MT@setupfont{%
762   \MT@check@font
763   \ifMT@inlist@
764   <debug>\MT@show@pdfannot2%
765   \else
766   \MT@vinfo{Setting up font `\'MT@@font\'\'on@line}%
```

Now we can begin setting up the font for all features that the current pdf<sub>T</sub><sub>E</sub><sub>X</sub> 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, since then, the order doesn't matter.

```
767 \MT@protrusion
768 \MT@expansion
769 }
```

Interword spacing and kerning (pdf<sub>T</sub><sub>E</sub><sub>X</sub> 1.40).

```
770 \MT@requires@pdftex6
771 {\g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}}\relax
```

Disable ligatures (pdf<sub>T</sub><sub>E</sub><sub>X</sub> 1.30).

```
772 \MT@requires@pdftex5
773 {\g@addto@macro\MT@setupfont\MT@noligatures}\relax
774 \g@addto@macro\MT@setupfont{%
```

Debugging.

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

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

```
776 \MT@register@font
777 \fi
778 \MT@gl@et\MT@font\@empty
779 }
```

`\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.

```
780 \MT@requires@pdftex7{
781 \let\MT@copy@font\relax
782 \def\MT@copy@font@{%
```

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

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

`\MT@font@orig` pdf<sub>T</sub><sub>E</sub><sub>X</sub> doesn't allow to copy a font that has already been copied and expanded/let-tered spaced. Hence, we have to get the original.

```

784 \expandafter\ifx\MT@font@copy\relax
785 \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
786 \expandafter\ifx\MT@font@orig\relax
787 \MT@exp@two@c\MT@gl@et\MT@font@orig\font@name
788 \else
789 \MT@exp@two@c\let\font@name\MT@font@orig
790 \fi
791 \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
792 <debug>\MT@dinfol{creating new copy: \MT@font@copy}%

```

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

```

793 \MT@map@clist@c\MT@active@features{%
794 \expandafter\ifx\csname MT@%nameuse{MT@abbr@##1}\endcsname\relax\else
795 \def\@tempa{##1}%
796 \expandafter\MT@map@tlist@c
797 \csname MT@##1@doc@contexts\endcsname
798 \MT@rem@from@list
799 \fi
800 }%
801 \fi
802 \MT@exp@two@c\let\MT@font\MT@font@copy

```

We only need the font identifier for letterspacing.

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

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

```

804 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
805 }

```

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

```

806 \def\MT@rem@from@list#1{%
807 \expandafter\ifx\csname MT@%tempa @#1font@list\endcsname\relax\else
808 \expandafter\MT@exp@one@c\expandafter\MT@rem@from@clist\expandafter
809 \MT@font \csname MT@%tempa @#1font@list\endcsname
810 \fi
811 }
812 }\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@family 813 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
\MT@series 814 \def\MT@encoding{#1}%
815 \def\MT@family {#2}%
\MT@shape 816 \def\MT@series {#3}%
\MT@size 817 \def\MT@shape {#4}%
818 \def\MT@size {#5}%
```

`\MT@familyalias` Alias family?

```
819 \MT@ifdefined@n@TF{MT@MT@family @alias}%
820 { \MT@let@cn\MT@familyalias{MT@MT@family @alias}}%
821 { \let\MT@familyalias\@empty}%
822 }
```

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

```
\MT@feat
\MT@maybe@do 823 \newif\ifMT@do
824 \def\MT@maybe@do#1{%
```

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

```
825 \expandafter\csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
```

Begin with setting micro-typography to true for this font. The `\MT@checklist@...` tests will set it to false if the property is not in the list. The first non-empty list that does not contain a match will stop us (except for font).

```
826 \MT@dotrue
827 \edef\@tempa{\csname MT@#1@setname\endcsname}%
828 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
829 \MT@ifdefined@n@TF{MT@checklist@#1}%
830 {\csname MT@checklist@#1\endcsname}%
831 {\MT@checklist@{#1}}}%
832 {#1}%
833 }%
834 \else
835 \MT@dofalse
836 \fi
837 \ifMT@do
```

`\MT@feat` stores the current feature.

```
838 \def\MT@feat{#1}%
839 \csname MT@set@#1@codes\endcsname
840 \else
841 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
842 \fi
843 }
```

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

```
844 \def\MT@checklist@#1#2{%
845 <!debug> \MT@ifdefined@n@T
846 <debug> \MT@ifdefined@n@TF
847 {MT@#2list@#1@\@tempa}{%
```

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

```
848 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
849 \csname MT@#1\expandafter\endcsname
850 \csname MT@#2list@#1@\@tempa\endcsname
```

```

851     \ifMT@inlist@
852 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#2}: #1 ` \@nameuse{MT@#1}' in list}%
853     \MT@dotrue
854     \else
855 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#2}: #1 ` \@nameuse{MT@#1}' not in list}%
856     \MT@dofalse
857     \expandafter\MT@clist@break
858     \fi
859 }%

```

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.

```

860 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#2}: #1 list empty}}%
861 }

```

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

```

862 \def\MT@checklist@family#1{%
863 <!debug> \MT@ifdefined@n@T
864 <debug> \MT@ifdefined@n@TF
865     {MT@#1list@family@ \@tempa}{%
866     \MT@exp@two@n\MT@in@clist
867     \MT@family{\csname MT@#1list@family@ \@tempa\endcsname}%
868     \ifMT@inlist@
869 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family ` \@nameuse{MT@family}' in list}%
870     \MT@dotrue
871     \else
872 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family ` \@nameuse{MT@family}' not in list}%
873     \MT@dofalse
874     \ifx\MT@familyalias\@empty \else
875     \MT@exp@two@n\MT@in@clist
876     \MT@familyalias{\csname MT@#1list@family@ \@tempa\endcsname}%
877     \ifMT@inlist@
878 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: alias ` \MT@familyalias' in list}%
879     \MT@dotrue
880 <debug> \else\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: alias ` \MT@familyalias' not in list}%
881     \fi
882     \fi
883     \fi
884     \ifMT@do \else
885     \expandafter\MT@clist@break
886     \fi
887 }%
888 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family list empty}}%
889 }

```

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

```

890 \def\MT@checklist@size#1{%
891 <!debug> \MT@ifdefined@n@T
892 <debug> \MT@ifdefined@n@TF
893     {MT@#1list@size@ \@tempa}{%
894     \expandafter\MT@in@rlist\csname MT@#1list@size@ \@tempa\endcsname
895     \ifMT@inlist@
896 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size ` \MT@size' in list}%
897     \MT@dotrue
898     \else
899 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size ` \MT@size' not in list}%
900     \MT@dofalse
901     \expandafter\MT@clist@break
902     \fi
903 }%
904 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size list empty}}%
905 }

```

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

```
906 \def\MT@checklist@font#1{%
907   <debug> \MT@ifdefined@n@T
908   <debug> \MT@ifdefined@n@TF
909   {MT#1list@font@tempa}{%
```

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

```
910   \edef\tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
911   \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
912   \tempb \csname MT#1list@font@tempa\endcsname
913   \ifMT@inlist@
914   <debug> \MT@dinfo@n1{1}{\@nameuse{MT@abbr@#1}: font '\MT@font' in list}%
915   \expandafter\MT@clist@break
916   \else
917   <debug> \MT@dinfo@n1{1}{\@nameuse{MT@abbr@#1}: font '\MT@font' not in list}%
918   \MT@dofalse
919   \fi
920   }%
921   <debug> {\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#1}: font list empty}}%
922 }
```

### 14.2.1 Protrusion

`\MT@protrusion` Set up for protrusion?

```
923 \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.

```
924 \def\MT@set@pr@codes{%
```

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

```
925   \MT@if@list@exists{%
926     \MT@get@font@dimen@six
927     \MT@get@opt
928     \MT@reset@pr@codes
```

Get the name of the inheritance list and parse it.

```
929   \MT@get@inh@list
```

Set an input encoding?

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

Load additional lists?

```
931   \MT@load@list\MT@pr@c@name
932   \MT@set@listname
```

Load the main list.

```
933   \MT@let@cn\tempc{MT@pr@c@\MT@pr@c@name}%
934   \expandafter\MT@pr@do\tempc,\relax,%
935   }\MT@reset@pr@codes
936 }
```

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

```
937 \def\MT@set@all@pr#1#2{%
938   <debug> \MT@dinfo@n1{3}{-- lp/rp: setting all to #1/#2}%
939   \let\MT@temp@empty
940   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lrcode\MT@font\@tempcnta=#1\relax}}%
941   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\lrcode\MT@font\@tempcnta=#2\relax}}%
942   \MT@do@font\MT@temp
```

```

943 }

\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.
944 \def\MT@reset@pr@codes@{\MT@set@all@pr@z@z@}
945 \let\MT@reset@pr@codes\relax

\MT@compensate@tr@    If the font is letterspaced, we have to add the letterspacing amount to the margin
\MT@compensate@tr@    kerns. This command will be activated in \MT@set@tr@codes.
946 \MT@requires@pdftex6{
947   \g@addto@macro\MT@set@pr@codes\MT@compensate@tr@
948   \let\MT@compensate@tr@relax
949   \def\MT@compensate@tr@{%
950     <debug>\MT@dinfo@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%
951     \MT@do@font{\MT@compensate@tr@@\lcode \MT@compensate@tr@@\rcode}%
952   }

\MT@compensate@tr@@
953   \def\MT@compensate@tr@@#1{%
954     #1\MT@font\@tempcnta=\numexpr#1\MT@font\@tempcnta + \MT@letterspace@/2\relax
955   }
956 }\relax

\MT@gobble@settings    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
\MT@get@font@dimen@six a bug).
957 \def\MT@get@font@dimen@six{%
958   \ifnum\fontdimen6\MT@font=\z@
959     \MT@warning@n1{%
960       Font '\MT@font' does not specify its\MessageBreak
961       \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
962       \nameuse{MT@abbr@\MT@feat} will not work with this font}%
963     \expandafter\MT@gobble@settings
964   \else
965     \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
966   \fi
967 }
968 \def\MT@gobble@settings#1\@tempc,\relax,{}

\MT@pr@do    Split up the values and set \lcode and \rcode.
969 \def\MT@pr@do#1,{%
970   \ifx\relax#1\@empty\else
971     \MT@pr@split #1==\relax
972     \expandafter\MT@pr@do
973   \fi
974 }

\MT@pr@split    The keyval package would remove spaces here, which we needn't do since
                \SetProtrusion ignores spaces in the protrusion list anyway.
975 \def\MT@pr@split#1=#2=#3\relax{%
976   \def\@tempa{#1}%
977   \ifx\@tempa\@empty \else
978     \MT@get@slot
979     \ifnum\MT@char > \m@ne
980       \MT@get@char@unit
981       \MT@pr@split@val#2\relax
982     \fi
983   \fi
984 }

```

`\MT@pr@split@val`

```

985 \def\MT@pr@split@val#1,#2\relax{%
986   \def\@tempb{#1}%
987   \MT@ifempty\@tempb\relax{%
988     \MT@scale@to@em
989     \lcode\MT@font\MT@char=\@tempcntb
990   <debug>\MT@info@n1{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char: [#1]}%
991   }%
992   \def\@tempb{#2}%
993   \MT@ifempty\@tempb\relax{%
994     \MT@scale@to@em
995     \rcode\MT@font\MT@char=\@tempcntb
996   <debug>\MT@info@n1{4}{;;; rp (\MT@char): \number\rcode\MT@font\MT@char: [#2]}%
997   }%

```

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

```

998 \MT@ifdefined@c@T\MT@pr@inh@name{%
999   \MT@ifdefined@nT\MT@inh@\MT@pr@inh@name @\MT@char @}%
1000   \expandafter\MT@map@tlist@c
1001   \csname MT@inh@\MT@pr@inh@name @\MT@char @\endcsname
1002   \MT@set@pr@heirs
1003 }%
1004 }%
1005 }

```

`\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`.

```

1006 \MT@maybe@etex{
1007   \MT@requires@pdfTeX{
1008     \def\MT@scale@to@em{%
1009       \@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`.

```

1010   \MT@scale@\@tempcntb \@tempb \MT@dimen@six
1011   \ifnum\@tempcntb=z@ \else
1012     \MT@scale@factor
1013   \fi
1014 }

```

`\MT@get@charwd`

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

```

1015 \def\MT@get@charwd{%
1016 X   \MT@count=\fontcharwd\MT@font\MT@char\relax
1017 Q   \setbox\z@=\hbox{\MT@font \char\MT@char}%
1018 Q   \MT@count=\wd\z@
1019   \ifnum\MT@count=z@ \MT@info@missing@char \fi
1020 }

```



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

```

1021 \MT@requires@pdfTeX6{
1022   \g@addto@macro\MT@get@charwd{%
1023     \MT@ifdefined@c@T\MT@letterspace@
1024     {\advance\MT@count-\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1025   }
1026   }\relax
1027 }{

```

No adjustment with versions 0.14f and 0.14g.

```

1028 \def\MT@scale@to@em{%
1029   \MT@count=\@tempb\relax
1030   \ifnum\MT@count=\z@ \else
1031     \MT@scale@factor
1032   \fi
1033 }

```

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

```

1034 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1035 }
1036 }

```

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

```

1037 \def\MT@get@font@dimen#1{%
1038   \ifnum\fontdimen#1\MT@font=\z@
1039     \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1040       \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1041       You should use a different 'unit' for \MT@curr@list@name}%
1042   \else
1043     \MT@count=\fontdimen#1\MT@font
1044   \fi
1045 }

```

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

```

1046 \MT@maybe@etex{
1047   \def\MT@info@missing@char{%
1048     \MT@info@n1{Character '\the\MT@toks'
1049 X     \iffontchar\MT@font\MT@char
1050       has a width of 0pt
1051 X     \else is missing\fi
1052 Q     \MessageBreak (it's probably missing)
1053     \MessageBreak in font '\MT@font'. \MessageBreak
1054     Ignoring protrusion settings for this character}%
1055   }
1056 }

```

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

```

1057 \def\MT@scale@factor{%
1058   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1059     \expandafter\MT@scale\expandafter \@tempcntb
1060     \csname MT@\MT@feat @factor@\endcsname \@m
1061   \fi
1062   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
1063     \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @max\endcsname
1064   \else
1065     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1066       \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @min\endcsname
1067     \fi

```

```

1068 \fi
1069 }

\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.

1070 \def\MT@warn@code@too@large#1{%
1071   \@tempcnta=#1\relax
1072   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1073     \expandafter\MT@scale\expandafter\@tempcnta\expandafter
1074     \@m \csname MT@\MT@feat @factor@\endcsname
1075   \fi
1076   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1077   \MT@warning@n1{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
1078     is too large for character\MessageBreak
1079     `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1080     Setting it to the maximum of \number\@tempcnta}%
1081   \@tempcntb=#1\relax
1082 }

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

1083 \def\MT@get@opt{%
1084   \MT@set@listname

\MT@pr@factor@            Apply a factor?

\MT@sp@factor@1085   \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@1086   \MT@let@nn{MT@\MT@feat @factor@}%
1087     {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1088     \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1089       \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1090   }{%
1091     \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1092   }%

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

\MT@kn@unit@1093   \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1094     \MT@let@nn{MT@\MT@feat @unit@}%
1095     {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1096     \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\@empty
1097       \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1098         relative to character widths}%
1099     \else
1100       \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname@m@ne
1101         \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1102           relative to width of space}%
1103       \fi
1104     \fi
1105   }{%
1106     \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
1107   }%

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

1108   \let\MT@get@char@unit\relax
1109   \let\MT@get@space@unit@gobble
1110   \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\@empty
1111     \let\MT@get@char@unit\MT@get@charwd

```

```

1112 \else
1113 \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\m@ne
1114 \let\MT@get@space@unit\MT@get@font@dimen
1115 \else
1116 \expandafter\MT@get@unit\csname MT@MT@feat @unit@\endcsname
1117 \fi
1118 \fi

```

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

```

1119 \MT@ifdefined@n@T{MT@MT@feat @c@\csname MT@MT@feat @c@name\endcsname @preset}{%
1120 \csname MT@preset@MT@feat\endcsname
1121 \MT@let@nc{MT@reset@MT@feat @codes}\relax
1122 }%
1123 }

```

\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.

```

1124 \def\MT@get@unit#1{%
1125 \expandafter\MT@get@unit@#1 e!\@nil
1126 \ifx\x\@empty\else\let#1\x\fi
1127 \@defaultunits\@tempdima#1 pt\relax\@nnil
1128 \ifdim\@tempdima=\z@
1129 \MT@warning@n1{%
1130 Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
1131 width. Setting factors of list ~\@nameuse{MT@MT@feat @c@name}'\MessageBreak
1132 relative to character widths instead}%
1133 \let#1\@empty
1134 \let\MT@get@char@unit\MT@get@charwd
1135 \else
1136 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
1137 to \the\@tempdima}%
1138 \MT@count=\@tempdima\relax
1139 \fi
1140 }
1141 \def\MT@get@unit@#1e#2#3\@nil{%
1142 \ifx\#3\\\let\x\@empty \else
1143 \if m#2%
1144 \edef\x{#1\fontdimen6\MT@font}%
1145 \else
1146 \if x#2%
1147 \edef\x{#1\fontdimen5\MT@font}%
1148 \fi
1149 \fi
1150 \fi
1151 }

```

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

```

1152 \def\MT@set@inputenc#1{%

```

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

```

1153 \def\MT@cat{#1}%
1154 \edef\@tempa{MT@MT@feat @#1@\csname MT@MT@feat @#1@name\endcsname @inputenc}%
1155 \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1156 }

```

\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.

```

1157 \MT@addto@setup{%

```

```

1158 \ifpackageloaded{inputenc}{
1159   \ifpackageafter{inputenc}{2006/02/22}{
1160     \def\MT@set@inputenc@{%
1161       \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1162       \MT@load@inputenc
1163     }
1164   }{
1165     \let\MT@set@inputenc@\MT@load@inputenc
1166   }
1167 }{
1168   \def\MT@set@inputenc@{%
1169     \MT@error{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1170       \MessageBreak package isn't loaded. Ignoring key}%
1171     {You must load the `inputenc' package before you can use the `inputenc' key.}%
1172   }
1173 }
1174 }

```

`\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.

```

1175 \def\MT@load@inputenc{%
1176   \MT@normal@catcodes
1177   <debug>\MT@info@nl{2}{-- heir of \MT@char: #1}%
1178   \inputencoding{\@nameuse{\@tempa}}%
1179 }

```

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

```

1180 \def\MT@set@pr@heirs#1{%
1181   \lcode\MT@font#1=\lcode\MT@font\MT@char
1182   \rcode\MT@font#1=\rcode\MT@font\MT@char
1183   <debug>\MT@info@nl{2}{-- heir of \MT@char: #1}%
1184   <debug>\MT@info@nl{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
1185   <debug>                                     \number\rcode\MT@font\MT@char}%
1186 }

```

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

```

\MT@preset@pr 1187 \def\MT@preset@pr{%
1188   \expandafter\expandafter\expandafter\MT@preset@pr@
1189   \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1190 }
1191 \def\MT@preset@pr@#1,#2\@nil{%
1192   \ifx\MT@pr@unit@\@empty
1193     \MT@warn@preset@twidth{pr}%
1194     \let\MT@preset@aux\MT@preset@aux@factor
1195   \else
1196     \def\MT@preset@aux{\MT@preset@aux@space2}%
1197   \fi
1198   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1199   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1200   \MT@set@all@pr@\@tempa\@tempb
1201 }

```

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

```

\MT@preset@aux@factor 1202 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 1203   \@tempcntb=#1\relax
1204   \MT@scale@factor
1205   \edef#2{\number\@tempcntb}%
1206 }
1207 \def\MT@preset@aux@space#1#2#3{%
1208   \def\@tempb{#2}%
1209   \MT@get@space@unit#1%
1210   \MT@scale@to@em

```

```

1211 \edef#3{\number\@tempcntb}%
1212 }

```

\MT@warn@preset@towidth

```

1213 \def\MT@warn@preset@towidth#1{%
1214 \MT@warning@nl{%
1215 Cannot preset characters relative to their widths\MessageBreak
1216 for \@nameuse{MT@abbr#1} list ``\@nameuse{MT@#1@c@name}'. Presetting them%
1217 \MessageBreak relative to lem instead}%
1218 }

```

### 14.2.2 Expansion

\MT@expansion Set up for expansion?

```

1219 \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).

```

1220 \def\MT@set@ex@codes@es{%
1221 \MT@if@list@exists{%
1222 \MT@get@ex@opt
1223 \MT@reset@ef@codes
1224 \MT@get@inh@list
1225 \MT@set@inputenc{c}%
1226 \MT@load@list\MT@ex@c@name
1227 \MT@set@listname
1228 \MT@let@cn\@tempc{MT@ex@c@\MT@ex@c@name}%
1229 \expandafter\MT@ex@do\@tempc,\relax,%
1230 \MT@expandfont
1231 }\relax
1232 }

```

\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.

```

1233 \newif\ifMT@nonselected
1234 \def\MT@set@ex@codes@n{%
1235 \MT@nonselectedtrue
1236 \MT@if@list@exists
1237 \MT@get@ex@opt
1238 {%
1239 \let\MT@stretch@\MT@stretch
1240 \let\MT@shrink@\MT@shrink
1241 \let\MT@step@\MT@step
1242 \let\MT@auto@\MT@auto
1243 \let\MT@ex@factor@\MT@ex@factor
1244}%
1245 \MT@reset@ef@codes
1246 \MT@expandfont
1247 \MT@nonselectedfalse
1248 }

```

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

```

1249 \let\MT@set@ex@codes\MT@set@ex@codes@n

```

`\MT@expandfont`      Expand the font.

```
1250 \def\MT@expandfont{%
1251   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1252 }
```

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

```
1253 \def\MT@set@all@ex#1{%
1254   <debug>\MT@dinfoln{3}{-- ex: setting all to \number#1}%
1255   \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1256 }
1257 \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.

```
1258 \MT@requires@pdftex4{
1259   \def\MT@reset@ef@codes{%
1260     \ifnum\MT@ex@factor@=\@m \else
1261       \MT@reset@ef@codes@
1262     \fi
1263   }
1264 }{
1265   \let\MT@reset@ef@codes\MT@reset@ef@codes@
1266 }
```

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

```
1267 \def\MT@ex@do#1,{%
1268   \ifx\relax#1\@empty \else
1269     \MT@ex@split #1=\relax
1270     \expandafter\MT@ex@do
1271   \fi
1272 }
```

`\MT@ex@split`

```
1273 \def\MT@ex@split#1=#2=#3\relax{%
1274   \def\@tempa{#1}%
1275   \ifx\@tempa\@empty \else
1276     \MT@get@slot
1277     \ifnum\MT@char > \m@ne
1278       \@tempcntb=#2\relax
```

Take an optional factor into account.

```
1279     \ifnum\MT@ex@factor@=\@m \else
1280       \MT@scale\@tempcntb \MT@ex@factor@ \@m
1281     \fi
1282     \ifnum\@tempcntb > \MT@ex@max
1283       \MT@warn@ex@too@large\MT@ex@max
1284     \else
1285       \ifnum\@tempcntb < \MT@ex@min
1286         \MT@warn@ex@too@large\MT@ex@min
1287       \fi
1288     \fi
1289     \efcode\MT@font\MT@char=\@tempcntb
1290 <debug>\MT@dinfoln{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#2]}%
```

Heirs, heirs, I love thy heirs.

```
1291 \MT@ifdefined@cT\MT@ex@inh@name{%
1292   \MT@ifdefined@nT{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1293     \expandafter\MT@map@tlist@c
1294     \csname MT@inh@\MT@ex@inh@name @\MT@char @\endcsname
1295     \MT@set@ex@heirs
1296   }%
```

```

1297     }%
1298     \fi
1299     \fi
1300 }

\MT@warn@ex@too@large

1301 \def\MT@warn@ex@too@large#1{%
1302   \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1303     character\MessageBreak ``the\MT@toks' in \MT@curr@list@name.\MessageBreak
1304     Setting it to the maximum of \number#1}%
1305   \@tempcntb=#1\relax
1306 }

\MT@get@ex@opt    Apply different values to this font?
\MT@ex@factor@
1307 \def\MT@get@ex@opt{%
1308   \MT@set@listname
1309   \MT@ifdefined@n@TF{MT@ex@cc@MT@ex@cc@name @factor}{%
1310     \MT@let@cn\MT@ex@factor@{MT@ex@cc@MT@ex@cc@name @factor}%
1311     \MT@step@
1312     \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor/1000}%
1313     }{%
1314       \let\MT@ex@factor@MT@ex@factor
1315     }%
1316     \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1317     \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1318     \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
1319     \def\@tempa{autoexpand}%
1320     \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1321     \MT@ifdefined@n@T{MT@ex@cc@MT@ex@cc@name @preset}{%
1322       \MT@preset@ex
1323       \let\MT@reset@ef@codes\relax
1324     }%
1325 }

\MT@get@ex@opt@

1325 \def\MT@get@ex@opt@#1#2{%
1326   \MT@ifdefined@n@TF{MT@ex@cc@MT@ex@cc@name @#1}{%
1327     \MT@let@nn{MT@#1@}{MT@ex@cc@MT@ex@cc@name @#1}%
1328     \MT@vinfo{... : #2}%
1329     }{%
1330     \MT@let@nn{MT@#1@}{MT@#1}%
1331     }%
1332 }

\MT@set@ex@heirs

1333 \def\MT@set@ex@heirs#1{%
1334   \efcode\MT@font#1=\efcode\MT@font\MT@char
1335   <debug>\MT@dinfo@n1{2}{-- heir of \MT@char: #1}%
1336   <debug>\MT@dinfo@n1{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1337 }

\MT@preset@ex

1338 \def\MT@preset@ex{%
1339   \@tempcntb=\csname MT@ex@cc@MT@ex@cc@name @preset\endcsname\relax
1340   \MT@scale@factor
1341   \MT@set@all@ex@tempcntb
1342 }

```

### 14.2.3 Interword Spacing (Glue)

\MT@spacing      Adjustment of interword spacing?

```
1343 \MT@requires@pdftex6{
```

```

1344 \def\MT@spacing{\MT@maybe@do{sp}}

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

1345 \def\MT@set@sp@codes{%
1346   \MT@if@list@exists{%
1347     \MT@get@font@dimen@six
1348     \MT@get@opt
1349     \MT@reset@sp@codes
1350     \MT@get@inh@list
1351     \MT@set@inputenc{c}%
1352     \MT@load@list\MT@sp@cc@name
1353     \MT@set@listname
1354     \MT@let@cn\@tempc\MT@sp@c@\MT@sp@c@name}%
1355     \expandafter\MT@sp@do\@tempc,\relax,%
1356   }\MT@reset@sp@codes
1357 }

\MT@sp@do

1358 \def\MT@sp@do#1,{%
1359   \ifx\relax#1\@empty \else
1360     \MT@sp@split #1==\relax
1361     \expandafter\MT@sp@do
1362   \fi
1363 }

\MT@sp@split

1364 \def\MT@sp@split#1=#2=#3\relax{%
1365   \def\@tempa{#1}%
1366   \ifx\@tempa\@empty \else
1367     \MT@get@slot
1368     \ifnum\MT@char > \m@ne
1369       \MT@get@char@unit
1370       \MT@sp@split@val#2\relax
1371     \fi
1372   \fi
1373 }

\MT@sp@split@val      If unit=space, \MT@get@space@unit will be defined to fetch the corresponding
                        fontdimen (2 for the first, 3 for the second and 4 for the third argument).

1374 \def\MT@sp@split@val#1,#2,#3\relax{%
1375   \def\@tempb{#1}%
1376   \MT@ifempty\@tempb\relax{%
1377     \MT@get@space@unit2%
1378     \MT@scale@to@em
1379     \knbscode\MT@font\MT@char=\@tempcntb
1380     <debug>\MT@debug{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1381   }%
1382   \def\@tempb{#2}%
1383   \MT@ifempty\@tempb\relax{%
1384     \MT@get@space@unit3%
1385     \MT@scale@to@em
1386     \stbscode\MT@font\MT@char=\@tempcntb
1387     <debug>\MT@debug{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1388   }%
1389   \def\@tempb{#3}%
1390   \MT@ifempty\@tempb\relax{%
1391     \MT@get@space@unit4%
1392     \MT@scale@to@em
1393     \shbscode\MT@font\MT@char=\@tempcntb
1394     <debug>\MT@debug{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1395   }%
1396   \MT@ifdefined@c@T\MT@sp@inh@name{%

```



```

1397 \MT@ifdefined@nT{MT@inh@MT@sp@inh@name @\MT@char @}{%
1398 \expandafter\MT@map@tlist@c
1399 \csname MT@inh@MT@sp@inh@name @\MT@char @\endcsname
1400 \MT@set@sp@heirs
1401 }%
1402 }%
1403 }

\MT@set@sp@heirs
1404 \def\MT@set@sp@heirs#1{%
1405 \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1406 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1407 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1408 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1409 <debug>\MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1410 <debug> \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1411 }

\MT@set@all@sp
\MT@reset@sp@codes 1412 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1413 <debug>\MT@info@n1{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1414 \let\MT@temp@empty
1415 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
1416 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1417 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1418 \MT@do@font\MT@temp
1419 }
1420 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1421 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1422 \def\MT@preset@sp{%
1423 \expandafter\expandafter\expandafter\MT@preset@sp@
1424 \csname MT@sp@c@MT@sp@c@name @preset\endcsname@nil
1425 }
1426 \def\MT@preset@sp@#1,#2,#3@nil{%
1427 \ifx\MT@sp@unit@empty
1428 \MT@warn@preset@to@width{sp}%
1429 \MT@ifempty{#1}{\let\@tempa@empty}{\MT@preset@aux@factor{#1}\@tempa}%
1430 \MT@ifempty{#2}{\let\@tempc@empty}{\MT@preset@aux@factor{#2}\@tempc}%
1431 \MT@ifempty{#3}{\let\@tempb@empty}{\MT@preset@aux@factor{#3}\@tempb}%
1432 \else
1433 \MT@ifempty{#1}{\let\@tempa@empty}{\MT@preset@aux@space2{#1}\@tempa}%
1434 \MT@ifempty{#2}{\let\@tempc@empty}{\MT@preset@aux@space3{#2}\@tempc}%
1435 \MT@ifempty{#3}{\let\@tempb@empty}{\MT@preset@aux@space4{#3}\@tempb}%
1436 \fi
1437 \MT@set@all@sp\@tempa\@tempc\@tempb
1438 }
1439 }\relax

```

#### 14.2.4 Additional Kerning

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

```

1440 \MT@requires@pdftex6{
1441 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

1442 \def\MT@set@kn@codes{%
1443 \MT@if@list@exists{%
1444 \MT@get@font@dimen@six
1445 \MT@get@opt

```

```

1446 \MT@reset@kn@codes
1447 \MT@get@inh@list
1448 \MT@set@inputenc{c}%
1449 \MT@load@list\MT@kn@cc@name
1450 \MT@set@listname
1451 \MT@let@cn\@tempc{MT@kn@cc\MT@kn@cc@name}%
1452 \expandafter\MT@kn@do\@tempc,\relax,%
1453 }\MT@reset@kn@codes
1454 }

```

\MT@kn@do

```

1455 \def\MT@kn@do#1,{%
1456 \ifx\relax#1\@empty \else
1457 \MT@kn@split #1==\relax
1458 \expandafter\MT@kn@do
1459 \fi
1460 }

```

\MT@kn@split

```

1461 \def\MT@kn@split#1=#2=#3\relax{%
1462 \def\@tempa{#1}%
1463 \ifx\@tempa\@empty \else
1464 \MT@get@slot
1465 \ifnum\MT@char > \m@ne
1466 \MT@get@char@unit
1467 \MT@kn@split@val#2\relax
1468 \fi
1469 \fi
1470 }

```

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

```

1471 \def\MT@kn@split@val#1,#2\relax{%
1472 \def\@tempb{#1}%
1473 \MT@ifempty\@tempb\relax{%
1474 \MT@get@space@unit2%
1475 \MT@scale@to@em
1476 \knbccode\MT@font\MT@char=\@tempcntb
1477 <debug>\MT@info@nl{4}{;;; knbc (\MT@char): \number\knbccode\MT@font\MT@char: [#1]}%
1478 }%
1479 \def\@tempb{#2}%
1480 \MT@ifempty\@tempb\relax{%
1481 \MT@get@space@unit2%
1482 \MT@scale@to@em
1483 \knaccode\MT@font\MT@char=\@tempcntb
1484 <debug>\MT@info@nl{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1485 }%
1486 \MT@ifdefined@c@T\MT@kn@inh@name{%
1487 \MT@ifdefined@n@T{MT@inh@MT@kn@inh@name @\MT@char @}{%
1488 \expandafter\MT@map@tlist@c
1489 \csname MT@inh@MT@kn@inh@name @\MT@char @\endcsname
1490 \MT@set@kn@heirs
1491 }%
1492 }%
1493 }

```

\MT@set@kn@heirs

```

1494 \def\MT@set@kn@heirs#1{%
1495 \knbccode\MT@font#1=\knbccode\MT@font\MT@char
1496 \knaccode\MT@font#1=\knaccode\MT@font\MT@char
1497 <debug>\MT@info@nl{2}{-- heir of \MT@char: #1}%
1498 <debug>\MT@info@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char/%
1499 <debug> \number\knaccode\MT@font\MT@char}%

```

```

1500 }

\MT@set@all@kn
\MT@reset@kn@codes 1501 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1502 <debug>\MT@info@n1{3}{-- knac/knbc: setting all to #1/#2}%
1503 \let\MT@temp\empty
1504 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbcode\MT@font\@tempcnta=#1\relax}}%
1505 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knaccode\MT@font\@tempcnta=#2\relax}}%
1506 \MT@do@font\MT@temp
1507 }
1508 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1509 \let\MT@reset@kn@codes\relax

\MT@preset@kn
\MT@preset@kn@ 1510 \def\MT@preset@kn{%
1511 \expandafter\expandafter\expandafter\MT@preset@kn@
1512 \csname MT@kn@c@MT@kn@c@name @preset\endcsname\@nil
1513 }
1514 \def\MT@preset@kn@#1,#2\@nil{%
1515 \ifx\MT@kn@unit@\empty
1516 \MT@warn@preset@tewidth{kn}%
1517 \let\MT@preset@aux\MT@preset@aux@factor
1518 \else
1519 \def\MT@preset@aux{\MT@preset@aux@space2}%
1520 \fi
1521 \MT@ifempty{#1}{\let\@tempa\empty}{\MT@preset@aux{#1}\@tempa}%
1522 \MT@ifempty{#2}{\let\@tempb\empty}{\MT@preset@aux{#2}\@tempb}%
1523 \MT@set@all@kn\@tempa\@tempb
1524 }
1525 }\relax

```

### 14.2.5 Tracking

This only works with pdfTeX 1.40.

```

1526 \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 1527 \let\MT@tr@font@list\empty
1528 \def\MT@tracking@{%
1529 \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
1530 \ifMT@inlist@else
1531 \MT@maybe@do{tr}%
1532 \ifMT@do@else
1533 \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1534 \fi
1535 \fi
1536 }
1537 \let\MT@tracking\MT@tracking@
1538 </package>
1539 <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.

1540 \def\MT@set@tr@codes{%
1541 <*package>
1542 \MT@vinfo{Tracking font ~\MT@font'\on@line}%
1543 \MT@get@font@dimen@six
1544 \MT@if@list@exists{%
1545 \MT@set@listname

```

```

1546 \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name}{%
1547 \MT@let@cn\MT@letterspace{MT@tr@cc@MT@tr@cc@name}%

\MT@tr@unit@    Different unit?

1548 \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name @unit}{%
1549 \MT@let@cn\MT@tr@unit@{MT@tr@cc@MT@tr@cc@name @unit}%
1550 \ifdim\MT@tr@unit@=1em
1551 \let\MT@tr@unit@\@undefined
1552 \else
1553 \MT@let@cn\@tempb{MT@tr@cc@MT@tr@cc@name}%
1554 \MT@get@unit\MT@tr@unit@
1555 \let\MT@tr@factor@\@em
1556 \MT@scale@to@em
1557 \edef\MT@letterspace{\number\@tempcntb}%
1558 \fi
1559 }%
1560 }%

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

1561 \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name @noligatures}%
1562 {\MT@let@cn\MT@tr@ligatures{MT@tr@cc@MT@tr@cc@name @noligatures}}%

\MT@tr@ispace    Adjust interword spacing.
\MT@tr@ospace 1563 \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name @spacing}%
1564 {\MT@let@cn\MT@tr@ispace{MT@tr@cc@MT@tr@cc@name @spacing}}%
1565 \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name @outerspacing}%
1566 {\MT@let@cn\MT@tr@ospace{MT@tr@cc@MT@tr@cc@name @outerspacing}}%
1567 }\relax

    Activate protrusion adjustment.
1568 \let\MT@compensate@tr\MT@compensate@tr@
1569 </package>
1570 \MT@ifdefined@c@TF\MT@letterspace@{\relax{\let\MT@letterspace@MT@letterspace}%
1571 \ifnum\MT@letterspace@=z@{\else
1572 <package> \MT@vinfo{... Tracking by \number\MT@letterspace@}%

    Letterspacing only works in PDF mode.
1573 \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 accur-
ately (taking substitutions into account), \font@name is guaranteed to correspond
to an actual font identifier.

1574 \xdef\MT@lsfont{\csname\expandafter\string\font@name
1575 \number\MT@letterspace@ls\endcsname}%
1576 \expandafter\ifx\MT@lsfont\relax
1577 <debug>\MT@info@nl{1}{... new letterspacing instance}%

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

1578 \MT@get@ls@basefont
1579 \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@

    Scale interword spacing (not configurable in letterspace).

1580 <*package>
1581 \MT@ifdefined@c@TF\MT@tr@ispace
1582 {\let\@tempa\MT@tr@ispace}
1583 {\edef\@tempa{\MT@letterspace@*,,}}%
1584 \MT@ifdefined@c@TF\MT@tr@ospace

```

```

1585      {\edef\@tempa{\@tempa,\MT@tr@ospace}}
1586      {\edef\@tempa{\@tempa,,}}}%
1587      \expandafter\MT@tr@adjust@space\@tempa,%
1588 </package>
1589 <*letterspace>
1590      \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1591                                     * \fontdimen2\MT@lsfont/1000\relax
1592 </letterspace>

```

Disable ligatures (not configurable in letterspace).

```

1593 <package>      \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1594 <letterspace>  \tagcode\MT@lsfont`f=\m@ne
1595      \fi

```

Finally, let the letterspaced font propagate.

```

1596      \aftergroup\MT@set@lsfont
1597 <*package>
1598      \let\MT@font\MT@lsfont

```

Adjust surrounding spacing.

```

1599      \ifx\MT@ls@adjust\relax
1600      \global\MT@outer@space=\csname MT@outer@space\MT@font\endcsname
1601      \MT@tr@outer@l
1602      \aftergroup\aftergroup\aftergroup\MT@tr@outer@r
1603      \fi
1604 </package>
1605      \fi
1606 }

```

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

```

1607 \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.

```

1608 \DeclareRobustCommand\lsstyle{%
1609 <package> \def\MT@feat{tr}%
1610 \let\MT@tracking\MT@set@tr@codes
1611 \selectfont

```

Possibly remove the first and last kerns (for `\textls*`). Use scaling to avoid a ‘Dimension too large’.

```

1612 \MT@lskern=\dimexpr\MT@ifdefined@c@TF\MT@letterspace@
1613                \MT@letterspace@\MT@letterspace sp
1614                * \fontdimen6\MT@lsfont/2000\relax
1615 \MT@ls@adjust
1616 }

```

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

```

1617 \DeclareRobustCommand\lslig[1]{%
1618 \kern\MT@lskern
1619 {\MT@get@ls@basefont\font@name #1}%
1620 \kern\MT@lskern
1621 }

```

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

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

```

1622 \def\MT@get@ls@basefont{%

```

```

1623 \escapechar\m@ne
1624 \edef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1625 \expandafter\ifx\MT@ls@basefont\relax
1626 \MT@exp@two@c\MT@gl@et\MT@ls@basefont\font@name
1627 \else
1628 <debug>\MT@dinfo@n1{1}{... fixing base font}%
1629 \MT@exp@two@c\let\font@name\MT@ls@basefont
1630 \fi
1631 }

```

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

```

1632 <*package>
1633 \MT@requires@pdftex7{
1634 \def\MT@tr@noligatures{%
1635 \ifx\MT@tr@ligatures\empty
1636 \MT@noligatures@\MT@lsfont\@undefined
1637 \else
1638 \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1639 \fi
1640 }
1641 }{
1642 \def\MT@tr@noligatures{%
1643 \MT@warning@n1{%
1644 Disabling selected ligatures is only possible since\MessageBreak
1645 pdftex 1.40.4. Disabling all ligatures instead}%
1646 \global\let\MT@tr@noligatures\relax
1647 }
1648 }

```

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

```
1649 \newskip\MT@outer@space
```

\MT@tr@adjust@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.

```

1650 \def\MT@tr@adjust@space#1,#2,#3,#4,#5,#6,{%
1651 \let\MT@temp\empty
1652 \MT@adjust@space@{#1}{#4}{2}\@empty
1653 \MT@adjust@space@{#2}{#5}{3}\@plus
1654 \MT@adjust@space@{#3}{#6}{4}\@minus
1655 \global\MT@let@nc\MT@outer@space\expandafter\string\MT@lsfont\MT@temp
1656 }

```

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

```

1657 \def\MT@adjust@space@#1#2#3#4{%
1658 \MT@ifempty{#2}{%
1659 \MT@ifempty{#1}{%
1660 \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
1661 }{%
1662 \MT@tr@get@space{#1}{#3}{1000}%
1663 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1664 \fontdimen#3\MT@lsfont=\@tempdima
1665 }%
1666 }{%
1667 \MT@tr@get@space{#2}{#3}{2000}%
1668 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1669 \MT@ifempty{#1}\relax{%
1670 \MT@tr@get@space{#1}{#3}{1000}%
1671 \fontdimen#3\MT@lsfont=\@tempdima
1672 }%

```

```
1673 }%
1674 }
```

`\MT@tr@get@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.

```
1675 \def\MT@tr@get@space#1#2#3{%
1676   \MT@test@ast#1*\@nil{%
1677     \MT@ifdefined@c@TF\MT@tr@unit@
1678     {\edef\@tempb{#1}\MT@scale@to@em}
1679     {\@tempcntb=#1\relax}%
1680     \@tempdima=\dimexpr \dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
1681     -\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).

```
1682   \ifnum#2=\tw@
1683     \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
1684     \fi
1685     \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
1686   }%
1687   \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
1688   \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
1689 }%
1690 <debug>\MT@dinfo2{... adjusting font dimen #2 (#1): \the\@tempdima}%
1691 }
```

`\MT@tr@outer@l` Recall the last skip (must really be a space, not just a marker).

```
1692 \def\MT@tr@outer@l{%
1693   \ifdim\lastskip>5sp
1694     \unskip \hskip\MT@outer@space
1695   \fi
1696 }
```

`\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 1697 \def\MT@tr@outer@r{%
1698   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
1699 }
1700 \def\MT@tr@outer@r@{%
1701   \def\MT@temp*#1{\hskip\MT@outer@space}%
1702   \ifcat\egroup\noexpand\MT@tr@outer@next
1703     \unkern\egroup
1704     \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\x=}%
1705   \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.

```
1706   \ifx\maybe@ic\MT@tr@outer@next
1707     \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\x=}%
1708   \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).

```
1709   \ifx\check@icr\MT@tr@outer@next
1710     \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\x=}%
1711   \else
1712     \ifx\@sptoken\MT@tr@outer@next
1713     \def\MT@temp* {\hskip\MT@outer@space}%

```

```

1714     \else
1715     \ifx~\MT@tr@outer@next
1716     \def\MT@temp*~{\nobreak\hskip\MT@outer@space}%
1717     \else
1718     \ifx\ \MT@tr@outer@next \else
1719     \ifx\space\MT@tr@outer@next \else
1720     \ifx\@xobeysp\MT@tr@outer@next \else
1721     \def\MT@temp*{\}%
1722     \let\MT@tr@outer@next\relax
1723     \fi\fi\fi\fi\fi\fi\fi\fi
1724     \MT@temp*%
1725 }

```

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

```

\MT@tr@outer@icr@ 1726 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
1727 \def\MT@tr@outer@icr@{\%
1728   \let\@let@token= \MT@tr@outer@next
1729   \maybe@ic@
1730 }

```

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

```

1731 }{
1732 \DeclareRobustCommand\lsstyle{%
1733   \MT@warning{Letterspacing only works with pdftex version 1.40\MessageBreak
1734   or newer. You might want to use the `soul' package\MessageBreak
1735   instead}%
1736   \MT@glet\lsstyle\relax
1737 }
1738 }
1739 </package>

```

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

```

1740 \DeclareRobustCommand\textls{%
1741   \hmode\bgroup
1742   \@ifstar{\let\MT@ls@adjust\MT@ls@adjust@\MT@textls}%
1743   {\let\MT@ls@adjust\relax \MT@textls}%
1744 }

```

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

```

\MT@letterspace@ 1745 \newcommand\MT@textls[2][\%
1746   \MT@ifempty{#1}%
1747   {\let\MT@letterspace@\@undefined}%
1748   {\KV@@sp@def\MT@letterspace@{#1}%
1749   \MT@ls@too@large\MT@letterspace@}%
1750   \lsstyle #2%
1751   \MT@ls@adjust
1752   \egroup
1753 }

```

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

```

1754 \def\MT@ls@too@large#1{%
1755   \ifnum#1>\MT@tr@max
1756   \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
1757   \let#1\MT@tr@max
1758   \else
1759   \ifnum#1<\MT@tr@min
1760   \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
1761   \let#1\MT@tr@min
1762   \fi
1763   \fi
1764 }

```



`\MT@lskern` This dimen is used for the starred version of `\textls` and for `\lslig`.

```
1765 \newdimen\MT@lskern
```

`\MT@lsadjust` Remove extra space for the starred version.

```
\MT@lsadjust@ 1766 \let\MT@lsadjust\relax
1767 \def\MT@lsadjust@{\kern-\MT@lskern}
1768 <package>
```

### 14.2.6 Disabling Ligatures

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

```
1769 \MT@requires@pdftex5{
1770   \def\MT@noligatures{%
1771     \MT@dotrue
1772     \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1773       \MT@ifdefined@n@TF{MT@checklist@##1}%
1774       {\csname MT@checklist@##1\endcsname}%
1775       {\MT@checklist@{##1}}}%
1776     {nl}}%
1777   }%
1778   \ifMT@do
1779     \MT@noligatures@\MT@font\MT@n@ligatures
1780   \fi
1781 }
```

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

```
1782 \def\MT@noligatures@#1#2{%
1783   \MT@ifdefined@c@TF#2{%
```

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

```
1784   \MT@ifdefined@c@TF\tagcode{%
1785     \MT@map@clist@c#2{%
1786       \def\@tempa{##1}\MT@get@slot
1787       \ifnum\MT@char > \m@ne
1788         \tagcode#1\MT@char=\m@ne\fi
1789     }%
1790     \MT@vinfo{... Disabling ligatures for characters: #2}%
1791   }{%
1792     \pdfnoligatures#1%
1793     \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
1794       know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
1795       the font instead}%
1796   }%
1797 }{%
1798   \pdfnoligatures#1%
1799   \MT@vinfo{... Disabling ligatures}%
1800 }%
1801 }
1802 }\relax
```

### 14.2.7 Loading the Configuration

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

```
1803 \def\MT@load@list#1{%
1804   \edef\@tempa{#1}%
1805   \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa @load}%
1806   \MT@ifstreq\@tempa\@tempb{%
1807     \MT@warning{\@nameuse{MT@abbr@\MT@feat} list '~\@tempa' cannot load itself}%
1808   }{%
```

```

1809 \ifx\@tempb\relax \else
1810 \MT@ifdefined@n@TF{MT@\MT@feat @c@\@tempb}{%
1811 \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list \@tempb'}%
1812 \begingroup
1813 \MT@load@list\@tempb
1814 \endgroup
1815 \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list
1816 \noexpand\MessageBreak\@tempb'}%
1817 \MT@let@cn@\tempc{MT@\MT@feat @c@\@tempb}%
1818 \expandafter\csname MT@\MT@feat @do\expandafter\endcsname\@tempc,\relax,%
1819 }{%
1820 \MT@warning{\@nameuse{MT@abbr@\MT@feat} list \@tempb' undefined.
1821 Cannot load\MessageBreak it from list \@tempa'}%
1822 }%
1823 \fi
1824 }%
1825 }

```

`\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.

```

1826 \let\MT@file@list@empty
1827 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

1828 \MT@in@clist{#1}\MT@file@list
1829 \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.

```

1830 \MT@begin@catcodes
1831 \let\MT@begin@catcodes\relax
1832 \let\MT@end@catcodes\relax
1833 \InputIfFileExists{mt-#1.cfg}{%
1834 \edef\MT@curr@file{mt-#1.cfg}%
1835 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1836 \MT@xadd\MT@file@list{#1,%}
1837 }{%
1838 \MT@get@basefamily#1\@empty\@empty\@empty\@nil
1839 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1840 \ifMT@inlist@
1841 \MT@xadd\MT@file@list{#1,%}
1842 \else
1843 \InputIfFileExists{mt-\@tempa.cfg}{%
1844 \edef\MT@curr@file{mt-\@tempa.cfg}%
1845 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1846 \MT@xadd\MT@file@list{\@tempa,#1,%}
1847 }{%
1848 \MT@vinfo{... No configuration file mt-#1.cfg}%
1849 \MT@xadd\MT@file@list{#1,%}
1850 }%
1851 \fi
1852 }%
1853 \endgroup
1854 \fi
1855 }

```

`\MT@normal@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.

```

1856 \def\MT@normal@catcodes{%
1857   \makeatletter
1858   \catcode`\^7%
1859   \catcode`\_9%
1860   \catcode`\^^I9%
1861   \catcode`\^^M9%
1862   \catcode`\z@
1863   \catcode`\{\@ne
1864   \catcode`\}\@tw@
1865   \catcode`\#6%
1866   \catcode`\%14%
1867   \MT@map@tlist@n{!\!"$%&'(\)*+,-./\:;<=>?[\]\_`{|~-}\@makeother
1868 }
```

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

```

1869 \def\MT@begin@catcodes{%
1870   \begingroup
1871   \MT@normal@catcodes
```

Inside the configuration files, we don’t have to bother about spaces.

```

1872   \let\KV@sp@def\def
1873 }
```

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

```

1874 \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.

```

1875 \def\MT@get@basefamily#1#2#3#4\@nil{%
1876   \ifx\@empty#4%
1877     \def\@tempa{#1#2#3}%
1878   \else
1879     \let\@tempa\@empty
1880     \edef\@tempb{#1#2#3#4}%
1881     \expandafter\MT@get@basefamily@\@tempb\@nil
1882   \fi
1883 }
```

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

```

1884 \def\MT@get@basefamily@#1#2\@nil{%
1885   \edef\@tempa{\@tempa#1}%
1886   \ifx\#2\expandafter\@gobble\else\expandafter\@firstofone\fi
1887   {\MT@in@tlist{#2}\MT@variants
1888    \ifMT@inlist\else\MT@get@basefamily@#2\@nil\fi}%
1889 }
```

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

```

\MT@get@listname
\MT@get@listname@ 1890 \def\MT@get@listname#1{%
1891   <debug>\MT@edinfo@n1{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
```

Table 3: Order for matching font attributes

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

```

1892 \let\MT@listname\@undefined
1893 \def\@tempb{#1}%
1894 \MT@map@tlist@c\MT@try@order\MT@get@listname@
1895 }
1896 \def\MT@get@listname@#1{%
1897 \expandafter\MT@next@listname#1%
1898 \ifx\MT@listname\@undefined \else
1899 \expandafter\MT@tlist@break
1900 \fi
1901 }

```

\MT@try@order      Beginning with version 1.7, we always check for the font size. Since the matching order has become more logical now, it can be described in words, so that we don't need table 3 in the documentation part any longer and can cast it off here.

```

1902 \def\MT@try@order{%
1903 {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
1904 {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
1905 }

```

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

```

1906 \def\MT@next@listname#1#2#3#4{%
1907 \edef\@tempa{\MT@encoding
1908 \ifnum#1=\@ne \MT@family\fi
1909 \ifnum#2=\@ne \MT@series\fi
1910 \ifnum#3=\@ne \MT@shape\fi
1911 \ifnum#4=\@ne *\fi
1912 \MT@context}%
1913 <debug>\MT@info@nl{1}{trying \@tempa}%
1914 \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
1915 \MT@next@listname@#4%
1916 }{%

```

Also try with an alias family.

```

1917 \ifnum#1=\@ne
1918 \ifx\MT@familyalias\@empty \else
1919 \edef\@tempa{\MT@encoding
1920 \MT@familyalias
1921 \ifnum#2=\@ne \MT@series\fi
1922 \ifnum#3=\@ne \MT@shape\fi
1923 \ifnum#4=\@ne *\fi
1924 \MT@context}%
1925 <debug>\MT@info@nl{1}{(alias) \@tempa}%
1926 \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
1927 \MT@next@listname@#4%
1928 }{%
1929 \fi
1930 \fi
1931 }%
1932 }

```

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

```

1933 \def\MT@next@listname@#1{%
1934   \ifnum#1=\@ne
1935     \expandafter\MT@in@rlist\csname MT@\@tempb @\@tempa @size\endcsname
1936     \ifMT@inlist@
1937       \let\MT@listname\MT@size@name
1938     \fi
1939   \else
1940     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
1941   \fi
1942 }
```

`\MT@if@list@exists`

```

\MT@context 1943 \def\MT@if@list@exists{%
1944   \MT@let@cn\MT@context{MT@\MT@feat @context}%
1945   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
1946   \MT@get@listname{\MT@feat @c}%
1947   \MT@ifdefined@c@TF\MT@listname{%
1948     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
1949     \ifMT@nonselected
1950       \MT@vinfo{... Applying non-selected expansion (list '\MT@listname')}%
1951     \else
1952       \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list '\MT@listname'}%
1953     \fi
1954     \@firstoftwo
1955   }{%
```

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

```

1956   \MT@let@cn{MT@\MT@feat @c@name}\@empty
```

Don't warn if `selected=false`.

```

1957   \ifMT@nonselected
1958     \MT@vinfo{... Applying non-selected expansion (no list)}%
1959   \else
```

Tracking doesn't require a list, either.

```

1960     \MT@ifstreq\MT@feat{tr}\relax{%
1961       \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
1962       for font\MessageBreak'\MT@font'%
1963       \ifx\MT@context\@empty\else\space(context: '\MT@context')\fi.
1964       Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
1965     }%
1966   \fi
1967   \@secondoftwo
1968 }%
1969 }
```

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

```

\MT@context 1970 \def\MT@get@inh@list{%
1971   \let\MT@context\@empty
1972   \MT@get@listname{\MT@feat @inh}%
1973   \MT@ifdefined@c@TF\MT@listname{%
1974     \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
1975   <debug>\MT@dinfo@n{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
1976   <debug>           '\MT@listname'}%
1977     \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%

```

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

```

1978   \ifx\@tempc\@empty \else
1979   <debug>\MT@dinfo@n{1}{parsing inheritance list ...}%

```

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

```

1980 \begingroup
1981 \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\MT@listname'}%
1982 \MT@set@inputenc{inh}%
1983 \expandafter\MT@inh@do\@tempc,\relax,%
1984 \global\MT@let@enc{MT@\MT@feat @inh@\MT@listname}\@empty
1985 \endgroup
1986 \fi
1987 }{%
1988 \MT@let@enc{MT@\MT@feat @inh@name}\@undefined
1989 }%
1990 }

```

### 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@ 1991 \def\MT@get@slot{%
1992 \escapechar~\
1993 \let\MT@char@m@ne
1994 \MT@norestrtrue

```

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

```

1995 \MT@toks=\expandafter{\@tempa}%

```

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

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

```

1996 \expandafter\MT@is@letter\@tempa\relax\relax
1997 \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.

```

1998 \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.

```

1999 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
2000 \MT@is@symbol

```

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

```

2001 {\expandafter\MT@is@composite\@tempa\relax\relax}%
2002 \ifnum\MT@char@ < \z@

```

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

```

2003      \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2004      \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2005      \fi
2006      \fi

2007      \let\MT@char\MT@char@
2008      \ifnum\MT@char < \z@
2009          \MT@warn@unknown
2010      \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.

```

2011      \ifMT@norest \else
2012          \MT@warn@rest
2013          \let\MT@char\m@ne
2014      \fi
2015      \fi
2016      \escapechar\m@ne
2017  }

```

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

```

2018 \newif\ifMT@norest

```

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

```

2019 \def\MT@is@letter#1#2\relax{%
2020     \ifcat a\noexpand#1\relax
2021         \edef\MT@char@{\number`#1}%
2022         \ifx\#2\%
2023             <debug>\MT@dinfoln{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2024             \else
2025                 \MT@norestfalse
2026             \fi
2027         \else
2028             \ifcat !\noexpand#1\relax
2029                 \edef\MT@char@{\number`#1}%
2030             <debug>\MT@dinfoln{3}{> `the\MT@toks' is a character (\MT@char@)}%
2031             \ifx\#2\%
2032                 \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
2033             \else
2034                 \MT@norestfalse
2035                 \expandafter\MT@is@number#1#2\relax\relax
2036             \fi
2037         \fi
2038     \fi
2039 }

```

`\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.

```

2040 \def\MT@is@number#1#2#3\relax{%
2041     \ifx\relax#3\relax \else
2042         \ifx\relax#2\relax \else
2043             \MT@noresttrue
2044             \if#1"\relax
2045                 \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2046             <debug>\MT@dinfoln{3}{> ... a hexadecimal number: \MT@char@}%

```

```

2047     \else
2048     \if#1'\relax
2049     \def\MT@char@{\number#1#2#3}%
2050 (debug)\MT@info@n1{3}{> ... an octal number: \MT@char@}%
2051     \else
2052     \MT@ifint{#1#2#3}{%
2053     \def\MT@char@{\number#1#2#3}%
2054 (debug)\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
2055     }\MT@noestfalse
2056     \fi
2057 \fi
2058 \ifnum\MT@char@ > \@cclv
2059 \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2060 \let\MT@char@\m@ne
2061 \fi
2062 \fi
2063 \fi
2064 }

```

`\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`) are also supported (however, not `ucs`, aka. `inputenc/utf8x`).

```

2065 \def\MT@is@active#1#2\@nil{%
2066 \ifnum\catcode`#1 = \active
2067 \begingroup
2068 \set@display@protect
2069 \let\IeC\@firstofone
2070 \let\@inpenc@undefined@\MT@undefined@char

```

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

```

2071 \def\UTFviii@defined##1{\ifx ##1\relax
2072 \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
2073 \edef\x{%
2074 \def\noexpand\@tempa{\@tempa}%

```

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

```

2075 \MT@toks={\the\MT@toks\space(= \@tempa)}%
2076 }%
2077 \expandafter\endgroup\x
2078 \fi
2079 }

```

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

```

2080 \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.

```

2081 \def\MT@is@symbol{%
2082 \expandafter\def\expandafter\MT@char\expandafter

```



```

2083     {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2084 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2085 \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2086 \ifnum\MT@char@ < \z@

```

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

```

2087 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2088 \fi
2089 }

```

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

```

\MT@charstring 2090 \begingroup
2091 \catcode`\=/\z@
2092 /MT@map@tlist@n{/CHAR}/@makeoother
2093 /lowercase{%
2094 /def/x{%
2095 /def/MT@charstring{\CHAR"%
2096 /def/MT@is@char##1\CHAR"##2##3##4/relax{%
2097 /ifx/relax##1/relax
2098 /if##3\relax
2099 /edef/MT@char@{/number"##2}%
2100 /MT@ifstreq/MT@charstring{##3##4}/relax/MT@noestfalse
2101 /else
2102 /edef/MT@char@{/number"##2##3}%
2103 /MT@ifstreq/MT@charstring{##4}/relax/MT@noestfalse
2104 /fi
2105 <debug> /MT@dinfo@n1{3}{> `the/MT@toks' is a \char (/MT@char@)}%
2106 /fi
2107 }%
2108 }%
2109 }
2110 /expandafter/endgroup/x

```

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

```

2111 \def\MT@is@composite#1#2\relax{%
2112 \ifx\#2\\\else

```

Again, we construct a control sequence, this time of the form: `\\<encoding>\<accent>-<character>`, e.g., `\\T1\"-a`, which we then expand once to see if it is a letter (if it has been defined by `\DeclareTextComposite`). This should be robust, finally, especially, since we also `\detokenize` the input instead of only `\stringifying` it. Thus, we will die gracefully even on wrong Unicode input without `utf8`.

```

2113 \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2114 \string\csname\MT@encoding\endcsname
2115 \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
2116 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2117 \fi
2118 }

```

(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 2119 \def\MT@set@listname{%
2120   \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list\noexpand\MessageBreak
2121   ~\@nameuse{MT@MT@feat @c@name}'}%
2122 }

```

`\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.

```

2123 \def\MT@warn@ascii{%
2124   \MT@warning@nl{Character ~\the\MT@toks' (= \MT@char@)
2125   is outside of ASCII range.\MessageBreak
2126   You must load the ~inputenc' package before using\MessageBreak
2127   8-bit characters in \MT@curr@list@name}%
2128 }

```

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

```

2129 \def\MT@warn@number@too@large#1{%
2130   \MT@warning@nl{%
2131     Number #1 in encoding ~\MT@encoding' too large!\MessageBreak
2132     Ignoring it in \MT@curr@list@name}%
2133 }

```

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

```

2134 \def\MT@warn@rest{%
2135   \MT@warning@nl{%
2136     Unknown slot number of character\MessageBreak~\the\MT@toks'%
2137     \MT@warn@maybe@inputenc\MessageBreak
2138     in font encoding ~\MT@encoding'.\MessageBreak
2139     Make sure it's a single character\MessageBreak
2140     (or a number) in \MT@curr@list@name}%
2141 }

```

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

```

2142 \def\MT@warn@unknown{%
2143   \MT@warning@nl{%
2144     Unknown slot number of character\MessageBreak~\the\MT@toks'%
2145     \MT@warn@maybe@inputenc\MessageBreak
2146     in font encoding ~\MT@encoding' in \MT@curr@list@name}%
2147 }

```

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

```

2148 \def\MT@warn@maybe@inputenc{%
2149   \MT@ifdefined@n@T
2150   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
2151   { (input encoding ~\@nameuse
2152   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2153 }

```

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

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

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

For my reference:

- `\pickup@font` is called by `\selectfont`, `\wrong@fontshape`, or `\getanddefine@font` (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 2154 \let\MT@font@list\empty
          2155 \let\MT@font\empty
```

`\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.

All this is done at the beginning of the document.

```
2156 </package>
2157 \MT@addto@setup{%
2158 <*package>
2159   \g@addto@macro\do@subst@correction
2160     {\xdef\MT@font{\csname \curr@fontshape/\f@size\endcsname}}
2161 </package>
```

`\MT@orig@pickupfont` Check whether `\pickup@font` is defined as expected. The warning issued by `\CheckCommand*` would be a bit too generic. microtype also seems to work with CJK.

```

2162 \ifpackageloaded{CJK}{
2163   \ifpackageafter{CJK}{2006/10/17}
2164   {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}} % 4.7.0
2165   {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}}
2166   \g@addto@macro\MT@orig@pickupfont{%
2167     {\expandafter\ifx\font@name\relax\define@newfont\fi}%
2168     {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2169       \define@newfont\else\xdef\font@name
2170         {\csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}
2171   }{
2172     \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
2173   }
2174   \ifx\pickup@font\MT@orig@pickupfont \else
2175     \MT@warning{\%
2176       Command \string\pickup@font\space is not defined as expected.%
2177     }*package
2178     \MessageBreak Double-check whether micro-typography is indeed\MessageBreak
2179     applied to the document.\MessageBreak (Hint: Turn on `verbose' mode)%
2180   }/package
2181 }
2182 \fi

```

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

```

2183 \g@addto@macro\pickup@font{\begingroup

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

```

```

2184 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
2185 \g@addto@macro\pickup@font{%
2186   \escapechar\m@ne
2187 }*package
2188 <debug> \global\MT@inannottrue
2189 <debug> \MT@glet\MT@pdf@annot\@empty
2190 <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.

```

2191 \ifx\MT@font\@empty
2192 \let\MT@font\font@name
2193 \else
2194 \ifx\MT@font\font@name \else
2195 <debug> \MT@addto@annot{= substituted with \MT@font}%
2196 \MT@register@subst@font
2197 \fi
2198 \fi
2199 \MT@setupfont
2200 }/package
2201 <letterspace> \MT@tracking
2202 \endgroup
2203 }
2204 }*package

```

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

```

2205 \let\MT@pickupfont\pickup@font

```

`\add@accent` Inside `\add@accent`, we have to disable microtype's setup, since the grouping in `\MT@orig@add@accent`

the patched `\pickup@font` would break the accent if different fonts are used for the base character and the accent. Fortunately,  $\text{\LaTeX}$  takes care that the fonts used for the `\accent` are already set up, so that we cannot be overlooking them.

```

2206 \let\MT@orig@add@accent\add@accent
2207 \def\add@accent#1#2{%
2208   \let\pickup@font\MT@orig@pickupfont
2209   \MT@orig@add@accent{#1}{#2}%
2210   \let\pickup@font\MT@pickupfont
2211 }
2212 </package>
2213 }
2214 <*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.

```

2215 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}

```

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

```

2216 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}

```

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

```

2217 \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.

```

2218 \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.

```

2219 \def\MT@check@font@cx{%
2220   \MT@if@true
2221   \MT@map@clist@c\MT@active@features{%
2222     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
2223     \MT@font \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2224     \ifMT@inlist@
2225       \MT@let@nc{\MT@\@nameuse{MT@abbr@##1}}\relax
2226     \else
2227       \MT@if@false
2228     \fi
2229   }%
2230   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2231 }

```

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

```

2232 \def\MT@register@subst@font@cx{%
2233   \MT@map@clist@c\MT@active@features{%
2234     \expandafter\MT@xadd
2235     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2236     {\font@name,}%
2237   }%
2238 }

```

<code>\MT@register@font@cx</code>	For each feature, add the current font to the list, unless we didn't set it up.
	<pre> 2239 \def\MT@register@font@cx{% 2240   \MT@map@clist@c\MT@active@features{% 2241     \expandafter\ifx\csname MT@\@nameuse{MT@abbr##1}\endcsname\relax\else 2242       \expandafter\MT@xadd 2243       \csname MT##1\csname MT##1@context\endcsname font@list\endcsname 2244       {\MT@font,}% 2245     \def\@tempa{##1}% 2246     \expandafter\MT@map@tlist@c 2247     \csname MT##1@doc@contexts\endcsname 2248     \MT@maybe@rem@from@list 2249   \fi 2250 }% 2251 } </pre>
<code>\MT@maybe@rem@from@list</code>	Recurse through all context font lists of the document and remove the font, unless it's the current context.
	<pre> 2252 \def\MT@maybe@rem@from@list#1{% 2253   \MT@ifstreql\@tempa/#1\{\@tempa/\csname MT@\@tempa @context\endcsname\}\relax{% 2254     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter 2255     \MT@font \csname MT@\@tempa @#1font@list\endcsname 2256   }% 2257 } </pre>
<code>\microtypecontext</code>	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.
	<pre> 2258 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}} 2259 \MT@addto@setup{% 2260   \def\microtypecontext#1{% 2261     \MT@setup@contexts 2262     \let\MT@reset@context\relax 2263     \setkeys{MTC}{#1}% 2264     \selectfont 2265     \MT@reset@context 2266   }% 2267 } </pre>
<code>\textmicrotypecontext</code>	This is just a wrapper around <code>\microtypecontext</code> .
	<pre> 2268 \def\textmicrotypecontext#1#2{{\microtypecontext{#1}#2}} </pre>
<code>\MT@reset@context</code>	We have to reset the font at the end of the group, provided there actually was a change.
<code>\MT@reset@context@</code>	<pre> 2269 \def\MT@reset@context@{% 2270   \MT@vinfo{&lt;&lt;&lt; Resetting contexts\on@line 2271   &lt;debug&gt; \MessageBreak= \MT@pr@context/\MT@ex@context 2272   &lt;debug&gt; /\MT@tr@context/\MT@kn@context/\MT@sp@context 2273   }% 2274   \selectfont 2275 } </pre>
<code>\MT@setup@contexts</code>	The first time <code>\microtypecontext</code> is called, we initialise the context lists and redefine the commands used in <code>\pickup@font</code> .
	<pre> 2276 \def\MT@setup@contexts{% 2277   \MT@map@clist@c\MT@active@features 2278   {\global\MT@let@nc\MT##1@font@list}\MT@font@list}% 2279   \MT@gl@et\MT@check@font\MT@check@font@cx 2280   \MT@gl@et\MT@register@font\MT@register@font@cx 2281   \MT@gl@et\MT@register@subst@font\MT@register@subst@font@cx 2282   \MT@gl@et\MT@setup@contexts\relax </pre>

2283 }

Define context keys.

```
2284 \MT@map@clist@c\MT@features@long{%
2285   \define@key{MTC}{#1}[]{}%
2286   \edef\@tempb{\@nameuse{MT@rbba#1}}%
2287   \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2288   \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).

```
2289   \MT@ifempty{#1}{\def\MT@val{}}{\def\MT@val{#1}}%
2290   \expandafter\ifx\csname MT@\@tempb @context\endcsname\MT@val
2291   <debug>\MT@dinfo{1}{>>> no change of #1 context: `\'MT@val'}%
2292   \else
2293     \MT@vinfo{>>> Changing #1 context to `\'MT@val'\MessageBreak\online
2294   <debug>   \space(previous: `\'@nameuse{MT@\@tempb @context}')}%
2295     }%
2296     \def\MT@reset@context{\aftergroup\MT@reset@context@}%
```

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

```
2297   \global\MT@let@nn\MT@reset@\@tempb @codes}\MT@reset@\@tempb @codes@}%
```

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

```
2298   \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
2299   \MT@val \csname MT@\@tempb @doc@contexts\endcsname
2300   \ifMT@inlist@ \else
2301     \expandafter\MT@xadd\csname MT@\@tempb @doc@contexts\endcsname{{\'MT@val}}%
2302   <debug> \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
2303   \fi
2304   \MT@edef@n{MT@\@tempb @context}{\'MT@val}%
2305   \fi
2306   \fi
2307   }%
2308 }
```

\MT@pr@context     Initialise the contexts.

```
\MT@ex@context 2309 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}%
\MT@tr@context 2310 \MT@def@n{MT@#1@context}{@}%
\MT@sp@context 2311 \MT@def@n{MT@#1@doc@contexts}{@}%
\MT@kn@context 2312 }
\MT@kn@context 2313 \let\MT@extra@context\@empty
```

\MT@pr@doc@contexts

\MT@ex@doc@contexts

\MT@tr@doc@contexts

\MT@sp@doc@contexts

\MT@kn@doc@contexts  
\DeclareMicrotypeSet  
\MT@extra@context  
\DeclareMicrotypeSet\*

## 14.3 Configuration

### 14.3.1 Font Sets

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

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

```
2314 \def\DeclareMicrotypeSet{%
2315   \ifstar
2316     \MT@DeclareSetAndUseIt
2317     \MT@DeclareSet
2318 }
```

\MT@DeclareSet

```

2319 \newcommand\MT@DeclareSet[3] [] {%
2320   \KV@sp@def\@tempa{#1}%
2321   \MT@ifempty\@tempa{%
2322     \MT@map@clist@c\MT@features{\MT@declare@sets{##1}{#2}{#3}}}%
2323   }%
2324   \MT@map@clist@c\@tempa{%
2325     \KV@sp@def\@tempa{##1}%
2326     \MT@ifempty\@tempa\relax{%
2327       \MT@ifisfeature{set declaration `#2'}{%
2328         \MT@exp@one@n\MT@declare@sets
2329         {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
2330       }%
2331     }%
2332   }%
2333 }%
2334 }

```

\MT@DeclareSetAndUseIt

```

2335 \newcommand\MT@DeclareSetAndUseIt[3] [] {%
2336   \MT@DeclareSet[#1]{#2}{#3}%
2337   \UseMicrotypeSet[#1]{#2}%
2338 }

```

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

```
2339 \let\MT@curr@set@name\@empty
```

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

```

2340 \def\MT@declare@sets#1#2#3{%
2341   \KV@sp@def\MT@curr@set@name{#2}%
2342   \MT@ifdefined@nT{MT@#1@set@@\MT@curr@set@name}{%
2343     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2344     \global\MT@let@nc{MT@#1list@size@\MT@curr@set@name}\@empty
2345   }%
2346   \global\MT@let@nc{MT@#1@set@@\MT@curr@set@name}\@empty
2347   <debug>\MT@info{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2348   \setkeys{MT@#1set}{#3}%
2349 }

```

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

```

2350 \def\MT@define@set@key@#1#2{%
2351   \define@key{MT@#2@set}{#1} [] {%
2352     \global\MT@let@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
2353     \MT@map@clist@n{##1}{%
2354       \KV@sp@def\MT@val{###1}%
2355       \MT@get@highlevel{#1}%

```

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

```

2356   \MT@exp@two@n@g@addto@macro
2357   {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
2358   {\MT@val,}%
2359 }%

```

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

```

2360   \expandafter\g@addto@macro\expandafter\MT@font@sets
2361   \csname MT@#2list@#1@\MT@curr@set@name\endcsname
2362   <debug>\MT@info@n1{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
2363 }%
2364 }

```

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



`\bfdefault` being expanded/protruded.

```
2365 \def\MT@get@highlevel#1{%
2366   \expandafter\MT@test@ast\MT@val*\@nil\relax{%
```

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

```
2367   \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
2368   \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.

```
2369 }%
2370 }
```

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

```
2371 \def\MT@test@ast#1*#2\@nil{%
2372   \def\@tempa{#1}%
2373   \MT@ifempty{#2}%
2374 }
```

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

```
\MT@fix@font@set 2375 \let\MT@font@sets\@empty
2376 \def\MT@fix@font@set#1{%
2377   \xdef#1{#1}%
2378   \global\@onelevel@sanitize#1%
2379 }
```

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

```
2380 \def\MT@define@set@key@size#1{%
2381   \define@key{MT@#1@set}{size}[]{%
2382     \MT@map@cliston{##1}{%
2383       \KV@esp@def\MT@val{###1}%
2384       \expandafter\MT@get@range\MT@val--\@nil
2385       \ifx\MT@val\relax \else
2386         \expandafter\MT@xadd
2387         \csname MT@#1list@size@\MT@curr@set@name\endcsname
2388         {{{\MT@lower}{\MT@upper}\relax}}}%
2389     \fi
2390   }%
2391   <debug>\MT@dinfol{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
2392   }%
2393 }
```

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 2394 \def\MT@get@range#1-#2-#3\@nil{%
2395   \MT@ifempty{#1}{%
2396     \MT@ifempty{#2}{%
2397       \let\MT@val\relax
2398     }%
2399     \def\MT@lower{0}%
2400     \def\MT@val{#2}%
2401     \MT@get@size
2402     \edef\MT@upper{\MT@val}%
```

```

2403 }%
2404 }{%
2405 \def\MT@val{#1}%
2406 \MT@get@size
2407 \ifx\MT@val\relax \else
2408 \edef\MT@lower{\MT@val}%
2409 \MT@ifempty{#2}{%
2410 \MT@ifempty{#3}%
2411 {\def\MT@upper{-1}}%

```

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

```

2412 {\def\MT@upper{2048}}%
2413 }{%
2414 \def\MT@val{#2}%
2415 \MT@get@size
2416 \ifx\MT@val\relax \else
2417 \MT@ifdim\MT@lower>\MT@val{%
2418 \MT@warning{%
2419 Invalid size range (\MT@lower\space > \MT@val) in font set
2420 ~\MT@curr@set@name'.\MessageBreak Swapping sizes}%
2421 \edef\MT@upper{\MT@lower}%
2422 \edef\MT@lower{\MT@val}%
2423 }{%
2424 \edef\MT@upper{\MT@val}%
2425 }%
2426 \MT@ifdim\MT@lower=\MT@upper
2427 {\def\MT@upper{-1}}%
2428 \relax
2429 \fi
2430 }%
2431 \fi
2432 }%
2433 }

```

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

```

2434 \def\MT@get@size{%

```

A single star would mean `\sizedefault`, which doesn't exist, so we define it to be `\normalsize`.

```

2435 \if*\MT@val\relax
2436 \def\@tempa{\normalsize}%
2437 \else
2438 \MT@let@cn\@tempa{\MT@val}%
2439 \fi
2440 \ifx\@tempa\relax \else

```

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

```

2441 \begingroup
2442 \def\set@fontsize##1##2##3##4\@nil{\gdef\MT@val{##2}}%
2443 \@tempa\@nil
2444 \endgroup
2445 \fi

```

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

```

2446 \MT@ifdimen\MT@val{%
2447 \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2448 \edef\MT@val{\strip@pt\@tempdima}%

```

```

2449 }{%
2450 \MT@warning{Could not parse font size `~\MT@val'\MessageBreak
2451           in font set `~\MT@curr@set@name'}%
2452 \let\MT@val\relax
2453 }%
2454 }

```

\MT@define@set@key@font

```

2455 \def\MT@define@set@key@font#1{%
2456 \define@key{MT@#1@set}{font}[]{%
2457   \global\MT@let@nc{MT@#1list@font@~\MT@curr@set@name}\@empty
2458   \MT@map@clist@n{##1}{%
2459     \KV@sp@def\MT@val{###1}%
2460     \expandafter\MT@get@font\MT@val///// \@nil
2461     \MT@exp@two@n@g@addto@macro
2462       {\csname MT@#1list@font@~\MT@curr@set@name\expandafter\endcsname}%
2463       {\MT@val,}%
2464   }%
2465   \expandafter\g@addto@macro\expandafter\MT@font@sets
2466     \csname MT@#1list@font@~\MT@curr@set@name\endcsname
2467   <debug>\MT@edinfo@n1{1}{-- font: \@nameuse{MT@#1list@font@~\MT@curr@set@name}}%
2468   }%
2469 }

```

\MT@get@font Translate any asterisks.

```

2470 \def\MT@get@font#1/#2/#3/#4/#5/#6@nil{%
2471 \MT@ifempty{#1#2#3#4#5}\relax{%
2472 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
2473 \ifx\MT@val\relax\def\MT@val{0}\fi
2474 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2475 \let\MT@val\@tempb
2476 }%
2477 }

```

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

```

2478 \def\MT@get@font@#1#2#3#4#5#6{%
2479 \let\@tempb\@empty
2480 \def\MT@temp{#1/#2/#3/#4/#5}%
2481 \MT@get@axis{encoding}{#1}%
2482 \MT@get@axis{family}{#2}%
2483 \MT@get@axis{series}{#3}%
2484 \MT@get@axis{shape}{#4}%
2485 \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
2486 \MT@ifempty{#5}{%
2487 \MT@warn@axis@empty{size}{\string\normalsize}%
2488 \def\MT@val{*}%
2489 }{%
2490 \def\MT@val{#5}%
2491 }%
2492 \MT@get@size
2493 }

```

\MT@get@axis

```

2494 \def\MT@get@axis#1#2{%
2495 \def\MT@val{#2}%
2496 \MT@get@highlevel{#1}%
2497 \MT@ifempty\MT@val{%
2498 \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2499 \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2500 }\relax
2501 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2502 }

```

\MT@warn@axis@empty

```
2503 \def\MT@warn@axis@empty#1#2{%
2504   \MT@warning{#1 axis is empty in font specification\MessageBreak
2505     `~\MT@temp'. Using `~#2' instead}%
2506 }
```

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

```
2507 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
2508   \MT@define@set@key@{encoding}{#1}%
2509   \MT@define@set@key@{family}{#1}%
2510   \MT@define@set@key@{series}{#1}%
2511   \MT@define@set@key@{shape}{#1}%
2512   \MT@define@set@key@size{#1}%
2513   \MT@define@set@key@font{#1}%
2514 }
```

\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.

```
2515 \renewcommand*\UseMicrotypeSet[2][]{%
2516   \KV@sp@def\@tempa{#1}%
2517   \MT@ifempty\@tempa{%
2518     \MT@map@clist@n\MT@features{{\MT@use@set{##1}{#2}}}%
2519   }{%
2520     \MT@map@clist@n\@tempa{%
2521       \KV@sp@def\@tempa{##1}%
2522       \MT@ifempty\@tempa\relax{%
2523         \MT@is@feature{activation of set `~#2'}{%
2524           \MT@exp@one@n\MT@use@set
2525             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2526         }%
2527       }%
2528     }%
2529   }%
2530 }
```

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

```
\MT@ex@setname 2531 \def\MT@use@set#1#2{%
\MT@tr@setname 2532   \KV@sp@def\@tempa{#2}%
2533   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@sp@setname 2534     \MT@xdef@n{MT@#1@setname}{\@tempa}%
\MT@kn@setname 2535     \MT@info{Using \@nameuse{MT@abbr@#1} set `~\@tempa'}%
2536   }{%
2537     \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2538       \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2539     }%
2540     \MT@warning{%
2541       The \@nameuse{MT@abbr@#1} set `~\@tempa' is undeclared.\MessageBreak
2542       Using set `~\@nameuse{MT@#1@setname}' instead}%
2543   }%
2544 }
```

\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.

```
2545 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2546   \KV@sp@def\@tempa{#1}%
2547   \MT@ifempty\@tempa{%
2548     \MT@map@clist@n\MT@features{{\MT@set@default@set{##1}{#2}}}%
2549   }{%
2550     \MT@map@clist@n\@tempa{%
2551       \KV@sp@def\@tempa{##1}%

```

```

2552     \MT@ifempty\@tempa\relax{%
2553         \MT@is@feature{declaration of default set `#2'}{%
2554             \MT@exp@one@n\MT@set@default@set
2555             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2556         }%
2557     }%
2558 }%
2559 }%
2560 }

\MT@default@pr@set
\MT@default@ex@set 2561 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 2562 \KV@sp@def\@tempa{#2}%
2563 \MT@ifdefined@n@TF{MT@#1@set@\@tempa}{%
\MT@default@sp@set 2564 <debug>\MT@info{1}{declaring default \@nameuse{MT@abbr@#1} set `@\@tempa'}%
\MT@default@kn@set 2565 \MT@xdef@n{MT@default@#1@set}{\@tempa}%
\MT@set@default@set 2566 }{%
2567     \MT@warning{%
2568         The \@nameuse{MT@abbr@#1} set `@\@tempa' is not declared.\MessageBreak
2569         Cannot make it the default set. Using set\MessageBreak `all' instead}%
2570     \MT@xdef@n{MT@default@#1@set}{all}%
2571 }%
2572 }

```

### 14.3.2 Variants and Aliases

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

```

\MT@variants
2573 \let\MT@variants\@empty
2574 \def\DeclareMicrotypeVariants{%
2575     \ifstar
2576         \MT@DeclareVariants
2577     {\let\MT@variants\@empty\MT@DeclareVariants}%
2578 }

```

`\MT@DeclareVariants`

```

2579 \def\MT@DeclareVariants#1{%
2580     \MT@map@clist@n{#1}{%
2581         \KV@sp@def\@tempa{##1}%
2582         \@onelevel@sanitize\@tempa
2583         \xdef\MT@variants{\MT@variants{\@tempa}}%
2584     }%
2585 }

```

`\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.

```

2586 \renewcommand*\DeclareMicrotypeAlias[2]{%
2587     \KV@sp@def\@tempa{#1}%
2588     \KV@sp@def\@tempb{#2}%
2589     \@onelevel@sanitize\@tempb
2590     \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2591         \MT@warning{Alias font family `@\@tempb' will override
2592             alias `@\@nameuse{MT@\@tempa @alias}'\MessageBreak
2593             for font family `@\@tempa'}}%
2594     \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.

```

2595 \MT@ifdefined@c@T\MT@family{%

```

```

2596 <debug>\MT@info{1}{Activating alias font ` \@tempb' for ` \MT@family'}%
2597 \MT@glet\MT@familyalias\@tempb
2598 }%
2599 }

```

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

```

2600 \def\LoadMicrotypeFile#1{%
2601 \KV@sp@def\@tempa{#1}%
2602 \@onelevel@sanitize\@tempa
2603 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2604 \ifMT@inlist@
2605 \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2606 \else
2607 \MT@xadd\MT@file@list{\@tempa,}%
2608 \MT@begin@catcodes
2609 \InputIfFileExists{mt-\@tempa.cfg}{%
2610 \edef\MT@curr@file{mt-\@tempa.cfg}%
2611 \MT@vinfo{... Loading configuration file \MT@curr@file}%
2612 }{%
2613 \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2614 does not exist}%
2615 }%
2616 \MT@end@catcodes
2617 \fi
2618 }

```

### 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 2619 \MT@requires@pdftex5{
2620 \def\DisableLigatures{%
2621 \MT@begin@catcodes
2622 \MT@DisableLigatures
2623 }
2624 \newcommand*\MT@DisableLigatures[2][{}]{%
2625 \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
2626 \xdef\MT@active@features{\MT@active@features,nl}%
2627 \global\MT@noligaturestrue
2628 \MT@declare@sets{nl}{no ligatures}{#2}%
2629 \gdef\MT@nl@setname{no ligatures}%
2630 \MT@end@catcodes
2631 }
2632 }{

```

If pdf<sub>T</sub><sub>E</sub>X is too old, we issue a warning.

```

2633 \renewcommand*\DisableLigatures[2][{}]{%
2634 \MT@warning{Disabling ligatures of a font is only possible\MessageBreak
2635 with pdftex version 1.30 or newer.\MessageBreak
2636 Ignoring \string\DisableLigatures}%
2637 }
2638 }

```

### 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.

```

2639 \def\DeclareMicrotypeBabelHook#1#2{%

```

```

2640 \MT@map@clist@n{#1}{%
2641 \KV@sp@def\@tempa{##1}%
2642 \MT@gdef@n{MT@babel\@tempa}{#2}%
2643 }%
2644 }

```

### 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).

```

2645 \def\SetProtrusion{%
2646 \MT@begin@catcodes
2647 \MT@SetProtrusion
2648 }

```

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

```
\MT@pr@c@name 2649 \newcommand*\MT@SetProtrusion[3] [] {%
```

```
\MT@extra@context 2650 \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.

```

2651 \MT@set@named@keys{MT@pr@c}{#1}%
2652 <debug>\MT@dinfo{1}{creating protrusion list `~\MT@pr@c@name'}%
2653 \def\MT@permutelist{pr@c}%
2654 \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>`, ...

```
2655 \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.

```

2656 \MT@gdef@n{MT@pr@c@MT@pr@c@name}{#3}%
2657 \MT@end@catcodes
2658 }

```

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

```

2659 \def\SetExpansion{%
2660 \MT@begin@catcodes
2661 \MT@SetExpansion
2662 }

```

`\MT@SetExpansion`

```
\MT@ex@c@name 2663 \newcommand*\MT@SetExpansion[3] [] {%
```

```
\MT@extra@context 2664 \let\MT@extra@context\@empty
```

```
\MT@permutelist 2665 \MT@set@named@keys{MT@ex@c}{#1}%
```

```
2666 \MT@ifdefined@n@T{MT@ex@c@MT@ex@c@name @factor}{%
```

```
2667 \ifnum\csize MT@ex@c@MT@ex@c@name @factor\endcsize > \@m
```

```
2668 \MT@warning@n1{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
```

```
2669 too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
```

```
2670 maximum of 1000}%
```

```
2671 \global\MT@let@nc{MT@ex@c@MT@ex@c@name @factor}\@m
```

```
2672 \fi
```

```

2673 }%
2674 <debug>\MT@info{1}{creating expansion list `~\MT@ex@c@name'}%
2675 \def\MT@permute\list{ex@c}%
2676 \setkeys{MT@cfg}{#2}%
2677 \MT@permute
2678 \MT@gdef@n{MT@ex@c@~\MT@ex@c@name}{#3}%
2679 \MT@end@catcodes
2680 }

```

\SetTracking

```

2681 \def\SetTracking{%
2682 \MT@begin@catcodes
2683 \MT@SetTracking
2684 }

```

\MT@SetTracking Third argument may be empty.

```

2685 \newcommand*\MT@SetTracking[3][]{%
2686 \let\MT@extra@context\empty
2687 \MT@set@named@keys{MT@tr@c}{#1}%
2688 <debug>\MT@info{1}{creating tracking list `~\MT@tr@c@name'}%
2689 \def\MT@permute\list{tr@c}%
2690 \setkeys{MT@cfg}{#2}%
2691 \MT@permute
2692 \MT@ifempty{#3}\relax{\MT@gdef@n{MT@tr@c@~\MT@tr@c@name}{#3}}%
2693 \MT@end@catcodes
2694 }

```

\SetExtraSpacing

```

2695 \def\SetExtraSpacing{%
2696 \MT@begin@catcodes
2697 \MT@SetExtraSpacing
2698 }

```

\MT@SetExtraSpacing

```

\MT@sp@c@name 2699 \newcommand*\MT@SetExtraSpacing[3][]{%
\MT@extra@context 2700 \let\MT@extra@context\empty
2701 \MT@set@named@keys{MT@sp@c}{#1}%
\MT@permute\list 2702 <debug>\MT@info{1}{creating spacing list `~\MT@sp@c@name'}%
2703 \def\MT@permute\list{sp@c}%
2704 \setkeys{MT@cfg}{#2}%
2705 \MT@permute
2706 \MT@gdef@n{MT@sp@c@~\MT@sp@c@name}{#3}%
2707 \MT@end@catcodes
2708 }

```

\SetExtraKerning

```

2709 \def\SetExtraKerning{%
2710 \MT@begin@catcodes
2711 \MT@SetExtraKerning
2712 }

```

\MT@SetExtraKerning

```

\MT@kn@c@name 2713 \newcommand*\MT@SetExtraKerning[3][]{%
\MT@extra@context 2714 \let\MT@extra@context\empty
2715 \MT@set@named@keys{MT@kn@c}{#1}%
\MT@permute\list 2716 <debug>\MT@info{1}{creating kerning list `~\MT@kn@c@name'}%
2717 \def\MT@permute\list{kn@c}%
2718 \setkeys{MT@cfg}{#2}%
2719 \MT@permute
2720 \MT@gdef@n{MT@kn@c@~\MT@kn@c@name}{#3}%
2721 \MT@end@catcodes
2722 }

```



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

```
2723 \def\MT@set@named@keys#1#2{%
2724   \def\x##1name=##2,##3\@nil{%
2725     \setkeys{#1}{name=##2}%
2726     \def\MT@options{##1##3}%
2727     \MT@rem@from@clist{name=}\MT@options
2728   }%
2729   \x#2,name=,\@nil
2730   \@expandtwoargs\setkeys{#1}\MT@options
2731 }
```

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

```
2732 \def\MT@define@code@key#1#2{%
2733   \define@key{MT@#2}{#1}[]{%
2734     \@tempcnta=\@ne
2735     \MT@map@clist@n{##1}{%
2736       \KV@sp@def\MT@val{###1}%
```

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

```
2737     \MT@get@highlevel{#1}%
2738     \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
2739     \advance\@tempcnta \@ne
2740   }%
2741 }%
2742 }
```

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

```
2743 \def\MT@define@code@key@size#1{%
2744   \define@key{MT@#1}{size}[]{%
2745     \MT@map@clist@n{##1}{%
2746       \KV@sp@def\MT@val{###1}%
2747       \expandafter\MT@get@range\MT@val--\@nil
2748       \ifx\MT@val\relax \else
2749         \expandafter\MT@xadd\csname MT@tempsize\endcsname
2750         {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2751       \fi
2752     }%
2753   }%
2754 }
```

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

```
2755 \def\MT@define@code@key@font#1{%
2756   \define@key{MT@#1}{font}[]{%
2757     \MT@map@clist@n{##1}{%
2758       \KV@sp@def\MT@val{###1}%
2759       \expandafter\MT@get@font@and@size\MT@val////\@nil
2760       \MT@xdef@n{MT@MT@permutelist @\@tempb\MT@extra@context}%
2761       {\csname MT@MT@permutelist @name\endcsname}%
2762       <debug>\MT@info@n{1}{initialising: use list for font \@tempb=\MT@val
2763       <debug>         \ifx\MT@extra@context\@empty\else\MessageBreak
2764       <debug>         (context: \MT@extra@context)\fi}%
2765       \expandafter\MT@xaddb
2766       \csname MT@MT@permutelist @\@tempb\MT@extra@context @sizes\endcsname
2767       {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
2768     }%
2769   }%
2770 }
```

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

```

2771 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2772   \MT@ifempty{#1#2#3#4#5}\relax
2773   {\MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}}%
2774 }

2775 \MT@define@code@key{encoding}{cfg}
2776 \MT@define@code@key{family}{cfg}
2777 \MT@define@code@key{series}{cfg}
2778 \MT@define@code@key{shape}{cfg}
2779 \MT@define@code@key@size{cfg}
2780 \MT@define@code@key@font{cfg}

```

\MT@define@opt@key

```

2781 \def\MT@define@opt@key#1#2{%
2782   \define@key{MT@#1c}{#2}[]{\MT@ifempty{##1}\relax%
2783     \MT@xdefn{MT@#1c@MT@curr@set@name @#2}{##1}}%
2784 }

```

The options in the optional first argument.

```

2785 \MT@map@clist@MT@features%

```

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

```

2786 \define@key{MT@#1c}{name}[]{%
2787   \MT@ifempty{##1}%
2788   \MT@edefn{MT@#1c@name}{\MT@curr@file/\the\inputlineno}%
2789   }{%
2790     \MT@edefn{MT@#1c@name}{##1}%
2791     \MT@ifdefined@n@T{MT@#1c@csname MT@#1c@name\endcsname}{%
2792       \MT@warning{Redefining \@nameuse{MT@abbr#1} list ~\@nameuse{MT@#1c@name}}%
2793       }%
2794     }%
2795     \MT@let@cn\MT@curr@set@name{MT@#1c@name}%
2796   }%
2797   \MT@define@opt@key{#1}{load}%
2798   \MT@define@opt@key{#1}{factor}%
2799   \MT@define@opt@key{#1}{preset}%
2800   \MT@define@opt@key{#1}{inputenc}%

```

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

```

2801 \define@key{MT@#1c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}%
2802 }

```

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

```

2803 \MT@requires@pdftex7{
2804   \define@key{MT@ex@c}{context}[]{\MT@ifempty{#1}\relax%
2805     \MT@gl@et\MT@copy@font\MT@copy@font@
2806     \def\MT@extra@context{#1}}%
2807   }
2808   \MT@addto@setup{%
2809     \define@key{MT@ex@c}{context}[]{%
2810       \ifx\MT@copy@font\MT@copy@font@
2811         \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
2812       \else
2813         \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
2814           Ignoring `context' key\on@line}%
2815         {Either move the settings inside the preamble,\MessageBreak
2816           or load the package with the `copyfonts' option.}%
2817       \fi
2818     }
2819   }

```

```

2820 }{
2821   \define@key{MT@ex@c}{context}[]{%
2822     \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
2823       or later. Ignoring `context' key\on@line}%
2824     {Upgrade pdftex.}%
2825   }
2826 }

```

\MT@warn@nodim

```

2827 \def\MT@warn@nodim#1{%
2828   \MT@warning{`@tempa' is not a dimension.\MessageBreak
2829     Ignoring it and setting values relative to\MessageBreak #1}%
2830 }

```

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

```

2831 \define@key{MT@pr@c}{unit}[character]{%
2832   \global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
2833   \def\@tempa{#1}%
2834   \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.

```

2835     \MT@ifdimen\@tempa
2836     {\global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
2837     {\MT@warn@nodim{character widths}}%
2838   }%
2839 }

```

Tracking may only be relative to a dimension.

```

2840 \define@key{MT@tr@c}{unit}[1em]{%
2841   \global\MT@let@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
2842   \def\@tempa{#1}%
2843   \MT@ifdimen\@tempa
2844   {\global\MT@let@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
2845   {\MT@warn@nodim{1em}%
2846     \MT@gdefn{MT@tr@c@MT@curr@set@name @unit}{1em}}%
2847 }

```

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

```

2848 \MT@map@clist@n{sp,kn}{%
2849   \define@key{MT@#1@c}{unit}[space]{%
2850     \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
2851     \def\@tempa{##1}%
2852     \MT@ifstreq\@tempa{character}\relax{%
2853       \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
2854       \MT@ifstreq\@tempa{space}\relax{%
2855         \MT@ifdimen\@tempa
2856         {\global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
2857         {\MT@warn@nodim{width of space}}%
2858       }%
2859     }%
2860   }%
2861 }

```

The first argument to \SetExpansion accepts some more options.

```

2862 \MT@map@clist@n{stretch,shrink,step}{%
2863   \define@key{MT@ex@c}{#1}[]{%
2864     \MT@ifempty{##1}\relax{%
2865       \MT@ifint{##1}{%

```

A space terminates the number.

```

2866         \MT@gdefn{MT@ex@c@MT@curr@set@name @#1}{##1 }%

```

```

2867     }{%
2868     \MT@warning{%
2869     Value `##1' for option `#1' is not a number.\MessageBreak
2870     Ignoring it}%
2871     }%
2872   }%
2873 }%
2874 }
2875 \define@key{MT@ex@ec}{auto}[true]{%
2876   \def\@tempa{#1}%
2877   \csname if\@tempa\endcsname

```

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

```

2878   \MT@requires@pdftex4{%
2879   \MT@gdefon{MT@ex@ec@MT@curr@set@name @auto}{autoexpand}%
2880   }{%
2881   \MT@warning{pdftex too old for automatic font expansion}%
2882   }
2883   \else
2884   \MT@requires@pdftex4{%
2885   \global\MT@letenc{MT@ex@ec@MT@curr@set@name @auto}\@empty
2886   }\relax
2887   \fi
2888 }

```

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

```

2889 \MT@define@opt@key{tr}{spacing}
2890 \MT@define@opt@key{tr}{outerspacing}
2891 \define@key{MT@tr@ec}{outer spacing}[]{\setkeys{MT@tr@ec}{outerspacing={#1}}}

```

Which ligatures should be disabled?

```

2892 \define@key{MT@tr@ec}{noligatures}[]%
2893   {\MT@xdefon{MT@tr@ec@MT@curr@set@name @noligatures}{#1}}
2894 \define@key{MT@tr@ec}{no ligatures}[]{\setkeys{MT@tr@ec}{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, \ r{a}, \ k{a}, \ u{a}), which will make the configuration files look much nicer and easier to maintain. If a single character of an inheritance list should have a different value, one can simply override it.

\MT@inh@feat      The optional argument may be used to restrict the list to some features,  
\MT@extra@inputenc      and to specify an input encoding.

```

2895 \renewcommand*\DeclareCharacterInheritance[1] [] {%
2896   \let\MT@extra@context\@empty
2897   \let\MT@extra@inputenc\@undefined
2898   \let\MT@inh@feat\@empty
2899   \setkeys{MT@inh@}{#1}%
2900   \MT@begin@catcodes
2901   \MT@set@inh@list
2902 }

```

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

```

2903 \def\MT@set@inh@list#1#2{%
2904   \MT@ifempty\MT@inh@feat{%

```

```

2905 \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
2906 }{%
2907 \MT@map@clist@c\MT@inh@feat{{%
2908 \KV@sp@def\@tempa{##1}%
2909 \MT@ifempty\@tempa\relax{%
2910 \MT@exp@one@n\MT@declare@char@inh
2911 {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
2912 }%
2913 }}%
2914 }%
2915 \MT@end@catcodes
2916 }

```

The keys for the optional argument.

```

2917 \MT@map@clist@c\MT@features@long{%
2918 \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}%
2919 \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}

```

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

```

2920 \def\MT@declare@char@inh#1#2#3{%
2921 \MT@edef@n{MT@#1@inh@name}%
2922 {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
2923 \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
2924 \MT@ifdefined@c@T\MT@extra@inputenc{%
2925 \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
2926 <debug>\MT@dinfo{1}{creating inheritance list ~\@nameuse{MT@#1@inh@name}}%
2927 \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
2928 \def\MT@permutelist{#1@inh}%
2929 \setkeys{MT@inh}{#2}%
2930 \MT@permute
2931 }

```

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

```

2932 \define@key{MT@inh}{encoding}[]{%
2933 \def\MT@val{#1}%
2934 \expandafter\MT@encoding@check\MT@val,\@nil
2935 \MT@get@highlevel{encoding}%
2936 \MT@edef@n{MT@tempencoding1}{\MT@val}%
2937 }

```

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

```

2938 \def\MT@encoding@check#1,#2\@nil{%
2939 \MT@ifempty{#2}\relax{%
2940 \edef\MT@val{#1}%
2941 \MT@warning{You may only specify one encoding for character\MessageBreak
2942 inheritance lists. Ignoring encoding(s) #2}%
2943 }%
2944 }

```

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

```

2945 \MT@define@code@key{family}{inh}
2946 \MT@define@code@key{series}{inh}
2947 \MT@define@code@key{shape}{inh}
2948 \MT@define@code@key{size}{inh}
2949 \MT@define@code@key{font}{inh}

```

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

```

\MT@set@<feature>@codes).
2950 \def\MT@inh@do#1,{%
2951   \ifx\relax#1\@empty \else
2952     \MT@inh@split #1==\relax
2953     \expandafter\MT@inh@do
2954   \fi
2955 }

\MT@inh@split    Only gather the inheriting characters here. Their codes will actually be set in
\MT@set@<feature>@codes.
2956 \def\MT@inh@split#1=#2=#3\relax{%
2957   \def\@tempa{#1}%
2958   \ifx\@tempa\@empty \else
2959     \MT@get@slot
2960     \ifnum\MT@char > \m@ne
2961       \let\MT@val\MT@char
2962       \MT@map@clist@n{#2}{%
2963         \def\@tempa{##1}%
2964         \ifx\@tempa\@empty \else
2965           \MT@get@slot
2966           \ifnum\MT@char > \m@ne
2967             \expandafter\MT@xadd
2968             \csname MT@inh@\MT@listname @\MT@val @\endcsname
2969             {{\MT@char}}%
2970           \fi
2971         \fi
2972       }%
2973   <debug>\MT@info@n{2}{children of #1 (\MT@val):
2974   <debug>          \@nameuse{MT@inh@\MT@listname @\MT@val @}}%
2975   \fi
2976 \fi
2977 }

```

### 14.3.7 Permutation

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

```

2978 \def\MT@permute{%
2979   \let\MT@cnt@encoding\@ne
2980   \MT@permute@

  Undefine commands for the next round.
2981   \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
2982   \MT@glet\MT@tempsize\@undefined
2983 }
2984 \def\MT@permute@{%
2985   \let\MT@cnt@family\@ne
2986   \MT@permute@@
2987   \MT@increment\MT@cnt@encoding
2988   \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
2989   \MT@permute@
2990 }
2991 \def\MT@permute@@{%
2992   \let\MT@cnt@series\@ne
2993   \MT@permute@@@
2994   \MT@increment\MT@cnt@family

```

```

2995 \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
2996 \MT@permute@@
2997 }
2998 \def\MT@permute@@@{%
2999 \let\MT@cnt@shape\@ne
3000 \MT@permute@@@
3001 \MT@increment\MT@cnt@series
3002 \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
3003 \MT@permute@@@
3004 }
3005 \def\MT@permute@@@@{%
3006 \MT@permute@@@@
3007 \MT@increment\MT@cnt@shape
3008 \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
3009 \MT@permute@@@@
3010 }

```

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

```

3011 \def\MT@permute@@@@@{%
3012 \MT@permute@define{encoding}%
3013 \ifMT@document
3014 \ifx\MT@tempencoding\@empty \else
3015 \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
3016 {\expandafter\expandafter\expandafter\@gobble}%
3017 \fi
3018 \fi
3019 \MT@permute@@@@@
3020 }

```

\MT@permute@@@@@

```

3021 \def\MT@permute@@@@@{%
3022 \MT@permute@define{family}%
3023 \MT@permute@define{series}%
3024 \MT@permute@define{shape}%
3025 \edef\@tempa{\MT@tempencoding
3026 \MT@tempfamily
3027 \MT@tempseries
3028 \MT@tempshape
3029 \MT@ifdefined@c@T\MT@tempsize *}%

```

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

```

3030 \def\@tempb{////}%
3031 \ifx\@tempa\@tempb \else
3032 \ifx\MT@tempencoding\@empty
3033 \MT@warning{%
3034 You have to specify an encoding for\MessageBreak
3035 \@nameuse{MT@abbr@MT@permutelist} list
3036 ~\@nameuse{MT@MT@permutelist @name}'.\MessageBreak
3037 Ignoring it}%
3038 \else
3039 \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

3040 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3041 \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3042 }%
3043 \expandafter\MT@xaddb
3044 \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
3045 \MT@tempsize
3046 <debug>\MT@info@n{1}{initialising: use list for font \@tempa,\MessageBreak

```

```

3047 <debug>          sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3048 <debug>          @sizes\endcsname}%
3049      }{%

```

Only one list can apply to a given combination.

```

3050      \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3051          \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3052              \@nameuse{MT@MT@permutelist @name}' will override list\MessageBreak
3053              \@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3054              for font \@tempa'}%
3055      }%
3056 <debug> \MT@info@n1{1}{initialising: use list for font \@tempa
3057 <debug>          \ifx\MT@extra@context\@empty\else\MessageBreak
3058 <debug>          (context: \MT@extra@context)\fi}%
3059      }%
3060      \MT@xdef@n{MT@MT@permutelist @\@tempa\MT@extra@context}%
3061      {\csname MT@MT@permutelist @name\endcsname}%
3062      \fi
3063      \fi
3064 }

```

\MT@permute@define     Define the commands.

```

3065 \def\MT@permute@define#1{%
3066     \expandafter\@tempcnta=\csname MT@cnt@#1\endcsname\relax
3067     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3068     {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3069     {\MT@let@nc{MT@temp#1}\@empty}%
3070 }

```

\MT@permute@reset     Reset the commands.

```

3071 \def\MT@permute@reset#1{%
3072     \@tempcnta=\@ne
3073     \MT@loop
3074         \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3075         \advance\@tempcnta\@ne
3076         \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3077         \iftrue
3078         \iffalse
3079     \MT@repeat
3080 }

```

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

```

3081 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

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

```

3082 \def\MT@check@rlist@#1#2#3{%
3083     \def\@tempb{#1}%
3084     \def\@tempc{#2}%
3085     \MT@if@false
3086     \expandafter\MT@map@tlist@c
3087     \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
3088     \MT@check@range
3089 }

```

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

```

3090 \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.

```

3091 \def\MT@check@range@#1#2#3{%

```



```

3092 \MT@ifdim{#2}=\m@ne{%
3093 \MT@ifdim\@tempc=\m@ne{%

```

- Both items are simple sizes.

```

3094 \MT@ifdim\@tempb={#1}\MT@iftrue\relax
3095 }{%

```

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

```

3096 \MT@ifdim\@tempb>{#1}\relax{%
3097 \MT@ifdim\@tempc>{#1}{%
3098 \MT@iftrue
3099 \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3100 }\relax
3101 }%
3102 }%
3103 }{%
3104 \MT@ifdim\@tempc=\m@ne{%

```

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

```

3105 \MT@ifdim\@tempb<{#2}{%
3106 \MT@ifdim\@tempb<{#1}\relax\MT@iftrue
3107 }\relax
3108 }{%

```

- Both items are ranges.

```

3109 \MT@ifdim\@tempb<{#2}{%
3110 \MT@ifdim\@tempc>{#1}{%
3111 \MT@iftrue
3112 \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3113 }\relax
3114 }\relax
3115 }%
3116 }%
3117 \ifMT@if@
3118 \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3119 \~@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
3120 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.

```

3121 \expandafter\MT@tlist@break
3122 \fi
3123 }

```

## 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 3124 \newif\ifMT@opt@expansion
```

```
\ifMT@opt@DVI 3125 \newif\ifMT@opt@auto
```

```
3126 \newif\ifMT@opt@DVI
```

`\MT@optwarn@admissible` Some warnings.

```

3127 \def\MT@optwarn@admissible#1#2{%
3128 \MT@warning@n1{~#1' is not an admissible value for option\MessageBreak
3129 ~#2'. Assuming ~false'}%
3130 }

```

\MT@optwarn@nan

```

3131 </package>
3132 \def\MT@optwarn@nan#1#2{%
3133   \MT@warning@n1{Value `#1' for option `#2' is not a\MessageBreak number.
3134               Using default value of \number\@nameuse{MT@#2@default}}%
3135 }
3136 <*package>

```

\MT@opt@def@set

```

3137 \def\MT@opt@def@set#1{%
3138   \MT@ifdefined@n@TF{MT@ \@tempb @set@{\MT@val}}{%
3139     \MT@xdef@n{MT@ \@tempb @setname}{\MT@val}%
3140   }{%
3141     \MT@xdef@n{MT@ \@tempb @setname}{\@nameuse{MT@default@ \@tempb @set}}%
3142     \MT@warning@n1{The #1 set `MT@val' is undeclared.\MessageBreak
3143               Using set ` \@nameuse{MT@ \@tempb @setname}' instead}%
3144   }%
3145 }

```

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

```

3146 \MT@map@clist@n{protrusion,expansion}{%
3147   \define@key{MT}{#1}[true]{%
3148     \csname MT@opt@#1true\endcsname
3149     \MT@map@clist@n{##1}{%
3150       \KV@sp@def\MT@val{###1}%
3151       \MT@ifempty\MT@val\relax{%
3152         \csname MT@#1true\endcsname
3153         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3154         \MT@ifstreq\MT@val{true}\relax
3155       }%
3156       \MT@ifstreq\MT@val{false}{%
3157         \csname MT@#1false\endcsname
3158       }{%
3159         \MT@ifstreq\MT@val{compatibility}{%
3160           \MT@let@nc{MT@ \@tempb @level}\@ne
3161         }{%
3162           \MT@ifstreq\MT@val{nocompatibility}{%
3163             \MT@let@nc{MT@ \@tempb @level}\tw@
3164           }{%

```

If everything failed, it should be a set name.

```

3165       \MT@opt@def@set{#1}%
3166     }%
3167   }%
3168 }%
3169 }%
3170 }%
3171 }%
3172 }%
3173 }

```

activate is a shortcut for protrusion and expansion.

```

3174 \define@key{MT}{activate}[true]{%
3175   \setkeys{MT}{protrusion={#1}}%
3176   \setkeys{MT}{expansion={#1}}%
3177 }

```

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

```

3178 \MT@map@clist@n{spacing,kerning,tracking}{%
3179   \define@key{MT}{#1}[true]{%
3180     \MT@map@clist@n{##1}{%

```

```

3181 \KV@sp@def\MT@val{###1}%
3182 \MT@ifempty\MT@val\relax{%
3183   \csname MT@#1true\endcsname
3184   \MT@ifstreq\MT@val{true}\relax
3185   {%
3186     \MT@ifstreq\MT@val{false}%
3187     \csname MT@#1false\endcsname
3188   }%
3189   \edef\@tempb{\csname MT@rbb@#1\endcsname}%
3190   \MT@opt@def@set{#1}%
3191 }%
3192 }%
3193 }%
3194 }%
3195 }%
3196 }

```

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

```

3197 \def\MT@def@bool@opt#1#2{%
3198   \define@key{MT}{#1}[true]{%
3199     \def\@tempa{##1}%
3200     \MT@ifstreq\@tempa{true}\relax{%
3201       \MT@ifstreq\@tempa{false}\relax{%
3202         \MT@optwarn@admissible{##1}{#1}%
3203         \def\@tempa{false}%
3204       }%
3205     }%
3206     #2%
3207   }%
3208 }

```

Boolean options that only set the switch.

```

3209 \MT@map@clist@n{draft,selected,babel}{%
3210   \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
3211 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotru}

```

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

```

3212 \MT@def@bool@opt{DVIoutput}{%
3213   \csname if\@tempa\endcsname
3214   \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
3215   \pdfoutput\z@
3216   \else
3217     \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
3218     \pdfoutput\@ne
3219   \fi
3220 }

```

Setting the deferssetup option to false will restore the old behaviour, where the setup took place at the time when the package was loaded. This is undocumented, since I would like to learn about the cases where this is necessary.

The only problem with the new deferred setup I can think of is when a box is being constructed inside the preamble and this box contains a font that is not loaded before the box is being used.

```

3221 \MT@def@bool@opt{deferssetup}{%
3222   \csname if\@tempa\endcsname \else
3223     \AtEndOfPackage{%
3224       \MT@setup@
3225       \let\MT@setup@\@empty

```

```

3226     \let\MT@addto@setup\@firstofone
3227   }%
3228   \fi
3229 }

```

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.

```

3230 \MT@requires@pdftex7{
3231   \MT@def@bool@opt{copyfonts}{%
3232     \csname if\@tempa\endcsname
3233     \MT@gllet\MT@copy@font\MT@copy@font@
3234     \else
3235     \MT@gllet\MT@copy@font\relax
3236   \fi
3237 }
3238 }{
3239   \MT@def@bool@opt{copyfonts}{%
3240     \csname if\@tempa\endcsname
3241     \MT@error{The pdftex version you are using is too old\MessageBreak
3242       to use the 'copyfonts' option}
3243     {Please upgrade pdftex.}%
3244   \fi
3245 }
3246 }

```

final is the opposite to draft.

```

3247 \MT@def@bool@opt{final}{%
3248   \csname if\@tempa\endcsname
3249   \MT@draftfalse
3250   \else
3251   \MT@drafttrue
3252   \fi
3253 }

```

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

```

3254 \define@key{MT}{verbose}[true]{%
3255   \let\MT@vinfo\MT@info@n1
3256   \def\@tempa{#1}%
3257   \MT@ifstreq\@tempa{true}\relax%

```

Take problems seriously.

```

3258   \MT@ifstreq\@tempa{errors}{%
3259     \let\MT@warning\MT@warn@err
3260     \let\MT@warning@n1\MT@warn@err
3261   }{%
3262     \let\MT@vinfo\@gobble
3263     \MT@ifstreq\@tempa{false}\relax
3264     {\MT@optwarn@admissible{#1}{verbose}}%
3265   }%
3266 }%
3267 }

```

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

```

3268 </package>
3269 \MT@map@clist@{%
3270 <package>   stretch,shrink,step,%
3271   letterspace}{%
3272   \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
3273     \def\@tempa{##1 }%

```

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

```

3274 \MT@ifint\@tempa
3275 {\MT@edef\MT@#1}{\@tempa}}%
3276 {\MT@optwarn@nan{##1}{#1}}%
3277 }%
3278 }
3279 *package

```

factor will define the protrusion factor only.

```

3280 \define@key{MT}{factor}[\MT@factor@default]{%
3281 \def\@tempa{#1}%
3282 \MT@ifint\@tempa
3283 {\edef\MT@pr@factor{\@tempa}}
3284 {\MT@optwarn@nan{#1}{factor}}%
3285 }

```

Unit for protrusion codes.

```

3286 \define@key{MT}{unit}[character]{%
3287 \def\@tempa{#1}%
3288 \MT@ifstreq\@tempa{character}\relax{%
3289 \MT@ifdimen\@tempa
3290 {\let\MT@pr@unit\@tempa}%
3291 {\MT@warning@n1{"\@tempa' is not a dimension.\MessageBreak
3292 Ignoring it and setting values relative to\MessageBreak
3293 character widths}}%
3294 }%
3295 }

```

#### 14.4.2 Reading the Configuration File

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as T<sub>E</sub>X systems are switching to the pdfT<sub>E</sub>X engine even for DVI output, so that the user might not even be aware of the fact that she's running pdfT<sub>E</sub>X.)

```

3296 \MT@protrusiontrue
3297 \ifnum\pdfoutput<\@ne \else

```

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

```

3298 \MT@requires@pdftex{
3299 \MT@expansiontrue
3300 \MT@autottrue
3301 }\relax
3302 \fi

```

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

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

```

3303 \define@key{MT}{config}[]{\relax}
3304 \def\MT@get@config#1config=#2,#3\@nil{%
3305 \MT@ifempty{#2}%
3306 {\def\MT@config@file{\MT@MT.cfg}}%
3307 {\def\MT@config@file{#2.cfg}}%
3308 }
3309 \expandafter\expandafter\expandafter\MT@get@config
3310 \curname opt@\currname.\@currentx\endcurname,config=\@nil

```

Load the file.

```

3311 \IfFileExists{\MT@config@file}{%

```

```

3312 \MT@info@n1{Loading configuration file \MT@config@file}%
3313 \MT@begin@catcodes
3314 \let\MT@begin@catcodes\relax
3315 \let\MT@end@catcodes\relax
3316 \let\MT@curr@file\MT@config@file
3317 \input{\MT@config@file}%
3318 \endgroup
3319 }{\MT@warning@n1{%
3320   Could not find configuration file '\MT@config@file'\MessageBreak
3321   This will almost certainly cause undesired results.\MessageBreak
3322   Please fix your installation}%
3323 }

```

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

```

3324 \MT@map@c1ist@c\MT@features{%
3325   \MT@ifdefined@n@TF{MT@default@#1@set}\relax
3326   {\MT@gdef@n{MT@default@#1@set}{all}}%
3327 }

```

`\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).

```

3328 \def\MT@check@active@set#1{%
3329   \MT@ifdefined@n@TF{MT@#1@setname}{%
3330     \MT@info@n1{Using \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}}%
3331   }{%
3332     \global\MT@let@nn{MT@#1@setname}{MT@default@#1@set}%
3333     \MT@info@n1{Using default \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}}%
3334   }%
3335 }

```

### 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.

```

3336 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
3337   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3338   Use \string\MicroType@Hook\space instead}\MicroType@Hook}
3339 \MT@ifdefined@c@T\MicroType@Hook\MicroType@Hook

```

#### 14.4.4 Changing Options Later

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

```

3340 \def\microtypesetup{\setkeys{MT}}
3341 \MT@addto@setup{\def\microtypesetup{\setkeys{MTX}}}
3342 \def\MT@define@optionX#1#2{%
3343   \define@key{MTX}{#1}[true]{%
3344     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3345     \MT@map@clist@n{##1}{%
3346       \KV@sp@def\MT@val{###1}%
3347       \MT@ifempty\MT@val\relax{%
3348         \@tempcnta=\m@ne
3349         \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.

```

3350   \MT@checksetup{#1}{%
3351     \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
3352     \MT@info{Enabling #1
3353       (level \number\csname MT@\@tempb @level\endcsname)}%
3354   }%
3355 }{%
3356   \MT@ifstreq\MT@val{false}{%
3357     \@tempcnta=\z@
3358     \MT@info{Disabling #1}%
3359   }{%
3360     \MT@ifstreq\MT@val{compatibility}{%
3361       \MT@checksetup{#1}{%
3362         \@tempcnta=\@ne
3363         \MT@let@nc{MT@\@tempb @level}\@ne
3364         \MT@info{Setting #1 to level 1}%
3365       }%
3366     }{%
3367       \MT@ifstreq\MT@val{nocompatibility}{%
3368         \MT@checksetup{#1}{%
3369           \@tempcnta=\tw@
3370           \MT@let@nc{MT@\@tempb @level}\tw@
3371           \MT@info{Setting #1 to level 2}%
3372         }%
3373       }{\MT@warning{%
3374         Value `~\MT@val' for key `~#1' not recognised.\MessageBreak
3375         Use any of `true', `false', `compatibility' or~\MessageBreak
3376         `nocompatibility'}%
3377     }%
3378   }%
3379 }%
3380 }%
3381 \ifnum\@tempcnta>\m@ne
3382   #2\@tempcnta\relax

```

```

3383     \fi
3384   }%
3385 }%
3386 }%
3387 }

```

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

```

3388 \def\MT@checksetup#1{%
3389   \expandafter\csname ifMT@#1\endcsname
3390   \expandafter\@firstofone
3391   \else
3392     \MT@warning{You cannot enable #1 if it was disabled\MessageBreak
3393               in the package options,}%
3394   \expandafter\@gobble
3395   \fi
3396 }

3397 \MT@define@optionX{protrusion}\pdfprotrudechars
3398 \MT@define@optionX{expansion}\pdfadjustspacing

```

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

```

3399 \MT@requires@pdftex6{
3400   \def\MT@define@optionX@#1#2{%
3401     \define@key{MTX}{#1}[true]{%
3402       \MT@map@clist@n{##1}{%
3403         \KV@sp@def\MT@val{###1}%
3404         \MT@ifempty\MT@val\relax{%
3405           \@tempcnta=\m@ne
3406           \MT@ifstreq\MT@val{true}{%
3407             \MT@checksetup{#1}{%
3408               \@tempcnta=\@ne
3409               \MT@info{Enabling #1}%
3410             }%
3411           }{%
3412             \MT@ifstreq\MT@val{false}{%
3413               \@tempcnta=\z@
3414               \MT@info{Disabling #1}%
3415             }{\MT@warning{Value `~\MT@val' for key `~#1' not recognised.
3416                       \MessageBreak Use either `true' or `false'}}%
3417           }%
3418         }%
3419         \ifnum\@tempcnta>\m@ne
3420           #2\relax
3421         \fi
3422       }%
3423     }%
3424   }%
3425 }

3426 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\relax\else
3427                               \let\MT@tracking\MT@tracking@\fi}
3428 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
3429 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3430                               \pdfappendkern \@tempcnta}
3431 }}

```

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

```

3432 \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
3433 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
3434 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
3435 }
3436 \define@key{MTX}{activate}[true]{%

```



```

3437 \setkeys{MTX}{protrusion={#1}}%
3438 \setkeys{MTX}{expansion={#1}}%
3439 }

```

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

```

3440 \let\MT@saved@setupfont\MT@setupfont
3441 \define@key{MTX}{disable}[]{%
3442   \MT@info{Inactivate `\'MT@MT' package}%
3443   \let\MT@setupfont\relax
3444 }
3445 \define@key{MTX}{enable}[]{%
3446   \MT@info{Reactivate `\'MT@MT' package}%
3447   \let\MT@setupfont\MT@saved@setupfont
3448 }

```

### 14.4.5 Processing the Options

\MT@ProcessOptionsWithKV      Parse options.

```

3449 </package>
3450 \def\MT@ProcessOptionsWithKV#1{%
3451   \let\@tempc\relax
3452   \let\MT@temp\@empty
3453   \MT@map@clist@c\@classoptionslist{%
3454     \def\CurrentOption{##1}%
3455     \MT@ifdefined@n@T{KV@#1@\CurrentOption}{%
3456       \edef\MT@temp{\MT@temp,\CurrentOption,%
3457         \@expandtwoargs\@removeelement\CurrentOption
3458         \@unusedoptionlist\@unusedoptionlist
3459       }%
3460     }%
3461     \edef\MT@temp{%
3462       \noexpand\setkeys{#1}{%
3463         \MT@temp\@optionlist{\@currname.\@current}%
3464       }%
3465     }%
3466     \MT@temp
3467     \MT@clear@options
3468   }
3469   \MT@ProcessOptionsWithKV{MT}
3470 <*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).

```

3471 \MT@addto@setup{%
3472   \ifMT@draft

```

We disable most of what we've just defined in the 3472 lines above if we are running in draft mode.

```

3473   \MT@warning@n1{\`draft' option active.\MessageBreak
3474     Disabling all micro-typographic extensions.\MessageBreak
3475     This might lead to different line and page breaks}
3476   \let\MT@setupfont\relax
3477   \renewcommand*\LoadMicrotypeFile[1]{}
3478   \renewcommand*\microtypesetup[1]{}
3479   \renewcommand*\microtypecontext[1]{}
3480   \renewcommand*\lsstyle{}
3481 \else

```

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

```
3482 \ifnum\pdfoutput<\@ne
3483   \ifMT@opt@expansion \else
3484     \MT@expansionfalse
3485   \fi
3486 \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!

```
3487 \MT@info@n1{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3488           \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
```

Fix the font sets.

```
3489 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
```

Protrusion.

```
3490 \ifMT@protrusion
3491   \edef\MT@active@features{\MT@active@features,pr}
3492   \pdfprotrudechars\MT@pr@level
3493   \MT@info@n1{Character protrusion enabled (level \number\MT@pr@level)%
3494     \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3495       factor: \number\MT@pr@factor\fi
3496     \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}
3497   \MT@check@active@set{pr}
3498 \else
3499   \let\MT@protrusion\relax
3500   \MT@info@n1{No character protrusion}
3501 \fi
```

Expansion.

```
3502 \ifMT@expansion
```

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

```
3503   \ifnum\MT@stretch=\m@ne
3504     \let\MT@stretch\MT@stretch@default
3505   \fi
```

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

```
3506   \ifnum\MT@shrink=\m@ne
3507     \ifnum\MT@stretch>\z@
3508       \let\MT@shrink\MT@stretch
3509     \else
3510       \let\MT@shrink\MT@shrink@default
3511     \fi
3512 \fi
```

If `step` has not been specified, we will set it to  $\min(\text{stretch}, \text{shrink})/5$ , rounded off, minimum value 1.

```
3513   \ifnum\MT@step=\m@ne
3514     \ifnum\MT@stretch>\MT@shrink
3515       \ifnum\MT@shrink>\z@
```

```

3516         \@tempcnta=\MT@stretch
3517     \else
3518         \@tempcnta=\MT@shrink
3519     \fi
3520 \else
3521     \ifnum\MT@stretch=\z@
3522         \@tempcnta=\MT@shrink
3523     \else
3524         \@tempcnta=\MT@stretch
3525     \fi
3526 \fi
3527 \divide\@tempcnta 5\relax
3528 \else
3529     \@tempcnta=\MT@step
3530     \ifnum\@tempcnta=\z@
3531         \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
3532             Setting it to one}
3533     \fi
3534 \fi
3535 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3536 \edef\MT@step{\number\@tempcnta\space}

```

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

```

3537     \let\MT@auto\@empty
3538     \ifMT@auto
3539         \MT@requires@pdftex4{%

```

We turn off automatic expansion if output mode is DVI.

```

3540         \ifnum\pdfoutput<\@ne
3541             \ifMT@opt@auto
3542                 \MT@warning@nl{%
3543                     Automatic font expansion only works for PDF output.\MessageBreak
3544                     However, you are creating a DVI file. I will switch\MessageBreak
3545                     automatic font expansion off and hope that expanded\MessageBreak
3546                     fonts are available}
3547             \fi
3548             \MT@autofalse
3549         \else
3550             \def\MT@auto{autoexpand}
3551         \fi

```

Also, if pdfTeX is too old.

```

3552     }{%
3553         \MT@warning@nl{%
3554             The pdftex version you are using is too old for\MessageBreak
3555             automatic font expansion. I will switch it off and\MessageBreak
3556             hope that expanded fonts are available.\MessageBreak
3557             Otherwise, install pdftex version 1.20 or newer}
3558         \MT@autofalse
3559         \def\MT@auto{1000 }
3560     }
3561 \else

```

No automatic expansion.

```

3562     \MT@requires@pdftex4\relax{
3563         \def\MT@auto{1000 }
3564     }
3565 \fi

```

Choose the appropriate macro for selected expansion.

```

3566 \ifMT@selected
3567 \let\MT@set@ex@codes\MT@set@ex@codes@s
3568 \else
3569 \let\MT@set@ex@codes\MT@set@ex@codes@n
3570 \fi

```

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

```

3571 \ifnum\MT@stretch=\z@
3572 \ifnum\MT@shrink=\z@
3573 \MT@warning@n1{%
3574 Both the stretch and shrink limit are set to zero.\MessageBreak
3575 Disabling font expansion}
3576 \MT@expansionfalse
3577 \fi
3578 \fi
3579 \fi
3580 \ifMT@expansion
3581 \edef\MT@active@features{\MT@active@features,ex}%
3582 \pdfadjustspacing\MT@ex@level
3583 \MT@info@n1{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
3584 (level \number\MT@ex@level),\MessageBreak
3585 stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3586 step: \number\MT@step, \ifMT@selected\else non-\fi selected}%
3587 \ifx\MT@copy@font\MT@copy@font@.\MessageBreak
3588 Set up for expansion contexts (using font copies)\fi}

```

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

```

3589 \def\MT@check@step#1{%
3590 \@tempcnta=\csname MT@#1\endcsname
3591 \divide\@tempcnta \MT@step
3592 \multiply\@tempcnta \MT@step
3593 \ifnum\@tempcnta=\csname MT@#1\endcsname\else
3594 \MT@warning@n1{The #1 amount is not a multiple of step.\MessageBreak
3595 The effective maximum #1 is \the\@tempcnta\space
3596 (step \number\MT@step)}
3597 \fi
3598 }
3599 \MT@check@step{stretch}
3600 \MT@check@step{shrink}
3601 \MT@check@active@set{ex}

```

Inside \showhyphens, font expansion should be disabled.

```

3602 \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3603 \color@begingroup\everypar{}\parfillskip\z@skip
3604 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3605 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

```

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

```

3606 \gdef\showhyphens#1{\setbox0\vbox{%
3607 \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3608 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3609 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
3610 \else
3611 \let\MT@expansion\relax
3612 \MT@info@n1{No font expansion}
3613 \fi
3614 }
3615 \MT@requires@pdftex6{
3616 </package>
3617 \MT@addto@setup{%
3618 <*package>

```

## Tracking, spacing and kerning.

```

3619 \ifMT@tracking
3620 <lua>
3621 \MT@requires@luatex{
3622 \MT@warning@nl{Tracking doesn't currently work with luatex}
3623 \let\MT@tracking\relax
3624 }{
3625 </lua>
3626 \edef\MT@active@features{\MT@active@features,tr}
3627 \MT@info@nl{Tracking enabled}
3628 \MT@check@active@set{tr}
3629 <lua>
3630 \else
3631 \let\MT@tracking\relax
3632 \MT@info@nl{No tracking}
3633 \fi
3634 \ifMT@spacing
3635 \edef\MT@active@features{\MT@active@features,sp}
3636 \pdfadjustinterwordglue\@ne
3637 \MT@info@nl{Adjustment of interword spacing enabled}
3638 \MT@check@active@set{sp}
3639 \else
3640 \let\MT@spacing\relax
3641 \MT@info@nl{No adjustment of interword spacing}
3642 \fi
3643 \ifMT@kerning
3644 \edef\MT@active@features{\MT@active@features,kn}
3645 \pdfprependkern\@ne
3646 \pdfappendkern\@ne
3647 \MT@info@nl{Adjustment of character kerning enabled}
3648 \MT@check@active@set{kn}
3649 \else
3650 \let\MT@kerning\relax
3651 \MT@info@nl{No adjustment of character kerning}
3652 \fi
3653 </package>

```

\MT@warn@tracking@DVI We issue a warning, when letterspacing in DVI mode, since it will probably not work.

```

3654 \ifnum\pdfoutput<\@ne
3655 \def\MT@warn@tracking@DVI{%
3656 \MT@warning@nl{%
3657 You are using tracking/letterspacing in DVI mode.\MessageBreak
3658 This will probably not work, unless the post-\MessageBreak
3659 processing program (dvips, dvipdfm(x), ...) is able\MessageBreak
3660 to create the virtual fonts on the fly}%
3661 \MT@glet\MT@warn@tracking@DVI\relax
3662 }
3663 \else
3664 \let\MT@warn@tracking@DVI\relax
3665 \fi
3666 \ifnum\MT@letterspace=\m@ne
3667 \let\MT@letterspace\MT@letterspace@default
3668 \else
3669 \MT@ls@too@large\MT@letterspace
3670 \fi
3671 }

```

If pdf<sub>T</sub>E<sub>X</sub> is too old, we disable tracking, spacing and kerning.

```

3672 <*package>
3673 }{
3674 \MT@addto@setup{%

```

```

3675 \ifMT@tracking
3676 \MT@warning@nl{Tracking only works with pdftex version 1.40\MessageBreak
3677 or newer. Switching it off}%
3678 \else
3679 \MT@info@nl{No tracking (pdftex too old)}
3680 \fi
3681 \ifMT@spacing
3682 \MT@warning@nl{Adjustment of interword spacing only works with\MessageBreak
3683 pdftex version 1.40 or newer. Switching it off}%
3684 \else
3685 \MT@info@nl{No adjustment of interword spacing (pdftex too old)}
3686 \fi
3687 \ifMT@kerning
3688 \MT@warning@nl{Character kerning only works with\MessageBreak
3689 pdftex version 1.40 or newer. Switching it off}%
3690 \else
3691 \MT@info@nl{No adjustment of character kerning (pdftex too old)}
3692 \fi
3693 }
3694 }

```

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>.

```

3695 \MT@requires@pdftex6{
3696 \AtBeginDocument{%
3697 \ifMT@spacing
3698 \ifMT@babel \else
3699 \ifnum\scode\>. > 1500
3700 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
3701 \MT@warning@nl{%
3702 \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3703 interword spacing will disable it. You might want\MessageBreak
3704 to add \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
3705 to your preamble}%
3706 }%
3707 \fi
3708 \fi
3709 \fi
3710 }
3711 }\relax

```

`\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```

3712 \MT@requires@pdftex5{
3713 \MT@addto@setup{%
3714 \ifMT@noligatures \else
3715 \let\MT@noligatures\relax
3716 \fi
3717 }
3718 }\relax

```

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

```

3719 \MT@addto@setup{%
3720 \ifx\MT@active@features\@empty \else
3721 \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3722 \fi
3723 \MT@documenttrue
3724 }

```

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

```

3725 \def\MT@set@babel@context#1{%
3726   \MT@ifdefined@n@TF{MT@babel@#1}{%
3727     \MT@info{*** Changing to language context `#1'\MessageBreak\on@line}%
3728     \expandafter\MT@exp@one@n\expandafter\microtypecontext
3729     \csname MT@babel@#1\endcsname
3730   }{%
3731     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3732   }%
3733 }
```

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

```

3734 \ifpackageloaded{babel}{
3735   \def\MT@shorthandoff#1#2{%
3736     \MT@info@n1{Switching off #1 babel's active characters}%
3737     \shorthandoff{#2}}
3738 }{
3739   \def\MT@shorthandoff#1#2{%
3740     \MT@error{You must load the `babel' package before \MT@MT}
3741     {Otherwise, `~\MT@MT' cannot switch off the active characters
3742      for #1 babel.}}
3743 }
```

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

```

3744 \MT@addto@setup{%
3745   \ifMT@babel
3746     \ifpackageloaded{babel}{%
3747       \MT@info@n1{Redefining babel's language switching commands}
3748       \let\MT@orig@select@language\select@language
3749       \def\select@language#1{%
3750         \MT@orig@select@language{#1}%
3751         \MT@set@babel@context{#1}%
3752       }
3753       \let\MT@orig@foreign@language\foreign@language
3754       \def\foreign@language#1{%
3755         \MT@orig@foreign@language{#1}%
3756         \MT@set@babel@context{#1}%
3757       }
3758     }
```

Disable French babel's active characters.

```

3758   \ifMT@kerning
3759     \MT@if@false
3760     \MT@with@babel@and@T{french} \MT@if@true
3761     \MT@with@babel@and@T{frenchb} \MT@if@true
3762     \MT@with@babel@and@T{francais} \MT@if@true
3763     \MT@with@babel@and@T{acadian} \MT@if@true
3764     \MT@with@babel@and@T{canadien} \MT@if@true
3765     \ifMT@if@MT@shorthandoff{French}{;!?}\fi
```

Disable Turkish babel's active characters.

```

3766     \MT@with@babel@and@T{turkish}{\MT@shorthandoff{Turkish}{:!=}}
3767     \fi
```

In case babel was loaded before microtype:

```

3768     \MT@set@babel@context\languagename
3769   }{%
3770     \MT@warning@n1{You did not load the babel package.\MessageBreak
3771     The `babel' option won't have any effect}
3772   }
3773   \fi
3774 }
```

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

```
3775 \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.

```
3776 \AtBeginDocument\selectfont
```

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

```
3777 \edef\MT@curr@file{\jobname.tex}
```

That was that.

```
3778 </package>
```

```
3779 </package|letterspace>
```

## 15 Configuration Files

Let's now write the font configuration files.

```
3780 <*config>
```

```
3781
```

### 15.1 Font Sets

We first declare some sets in the main configuration file.

```
3782 <*m-t>
```

```
3783 %%% -----
```

```
3784 %%% FONT SETS
```

```
3785
```

```
3786 \DeclareMicrotypeSet{all}
```

```
3787 { }
```

```
3788
```

```
3789 \DeclareMicrotypeSet{allmath}
```

```
3790 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
```

```
3791
```

```
3792 \DeclareMicrotypeSet{alltext}
```

```
3793 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
```

```
3794
```

```
3795 \DeclareMicrotypeSet{basicmath}
```

```
3796 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
```

```
3797   family   = {rm*,sf*},
```

```
3798   series   = {md*},
```

```
3799   size     = {normalsize,footnotesize,small,large}
```

```
3800 }
```

```
3801
```

```
3802 \DeclareMicrotypeSet{basictext}
```

```
3803 { encoding = {OT1,T1,LY1,OT4,QX,T5},
```

```
3804   family   = {rm*,sf*},
```

```
3805   series   = {md*},
```

```
3806   size     = {normalsize,footnotesize,small,large}
```

```
3807 }
```

```
3808
```

```
3809 \DeclareMicrotypeSet{smallcaps}
```

```
3810 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
```

```
3811   shape    = {sc}
```

```
3812 }
```

```
3813
```

```
3814 \DeclareMicrotypeSet{footnotesize}
```

```
3815 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
```

```
3816   size     = {-small}
```



```

3817 }
3818
3819 \DeclareMicrotypeSet{scriptsize}
3820 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3821   size      = {-footnotesize}
3822 }
3823
3824 \DeclareMicrotypeSet{normal font}
3825 { font = */*/*/*/ }
3826

```

The default sets.

```

3827 %%% -----
3828 %%% DEFAULT SETS
3829
3830 \DeclareMicrotypeSetDefault[protrusion]{alltext}
3831 \DeclareMicrotypeSetDefault[expansion]{basictext}
3832 \DeclareMicrotypeSetDefault[spacing]{basictext}
3833 \DeclareMicrotypeSetDefault[kerning]{alltext}
3834 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
3835

```

## 15.2 Font Variants and Aliases

```

3836 %%% -----
3837 %%% FONT VARIANTS AND ALIASES
3838

```

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

```

3839 \DeclareMicrotypeVariants{x,j,w,a,d,0,1}
3840

```

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.

```

3841 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
3842 \DeclareMicrotypeAlias{aer}{cmr} % ae
3843 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
3844 \DeclareMicrotypeAlias{cmor}{cmr} % eco
3845 \DeclareMicrotypeAlias{hfor}{cmr} % hfoldsty

```

The packages `pxfonts` and `txfonts` fonts inherit Palatino and Times settings respectively, also the  $\TeX$  Gyre fonts `Pagella` and `Termes` (formerly: `qfonts`).

```

3846 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
3847 \DeclareMicrotypeAlias{qpl}{ppl} % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

```

The 'FPL Neu' fonts, a 're-implementation' of Palatino.

```

3848 \DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
3849 \DeclareMicrotypeAlias{fp9j}{pplj} % "
3850 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
3851 \DeclareMicrotypeAlias{qtm}{ptm} % TeX Gyre Termes (formerly: qfonts/QuasiTimes)

```

More Times variants, to be checked: pns, mns (TimesNewRomanPS); mnt (TimesNewRomanMT), mntx (TimesNRExpertMT); mtm (TimesSmallTextMT); pte (Times-Europa); ptt, pttj (TimesTen); TimesEighteen; TimesModernEF.

The eulervm package virtually extends the Euler fonts.

```
3852 \DeclareMicrotypeAlias{eur}{eur} % Euler VM
3853 \DeclareMicrotypeAlias{zeus}{eus} % "
```

MicroPress's Charter version (chmath).

```
3854 \DeclareMicrotypeAlias{chr}{bch} % CH Math
```

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```
3855 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
3856 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
3857 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

Euro symbol fonts, to save some files.

```
3858 \DeclareMicrotypeAlias{zpeu}{zpeu} % Adobe Euro sans -> serif
3859 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
3860 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
3861
```

## 15.3 Interaction with babel

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

```
3862 %%% -----
3863 %%% INTERACTION WITH THE `babel' PACKAGE
3864
3865 \DeclareMicrotypeBabelHook
3866 {french,français,acadian,canadien}
3867 {kerning=french, spacing=}
3868
3869 \DeclareMicrotypeBabelHook
3870 {english,UKenglish,british,USenglish,american}
3871 {kerning=, spacing=nonfrench}
3872
3873 \DeclareMicrotypeBabelHook
3874 {turkish}
3875 {kerning=turkish, spacing=}
3876
```

## 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.

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.

```

3877 </m-t>
3878 <*m-t |zpeu|mv>
3879 %%%
3880 %%% CHARACTER INHERITANCE
3881
3882 </m-t |zpeu|mv>
3883 <*m-t>

```

```
3884 \DeclareCharacterInheritance
3885 { encoding = OT1 }
3886 { f = {011}, % ff
3887   i = {\i},
3888   j = {\j},
3889   O = {\O},
3890   o = {\o}
3891 }
3892
```

```

3893 \DeclareCharacterInheritance
3894 { encoding = T1 }
3895 { A = {\`A,\`A,\^A,\-A,\`A,\r A,\k A,\u A},
3896   a = {\`a,\`a,\^a,\-a,\`a,\r a,\k a,\u a},
3897   C = {\`C,\c C,\v C},
3898   c = {\`c,\c c,\v c},
3899   D = {\v D,\DH},
3900   d = {\v d,\dj},
3901   E = {\`E,\`E,\^E,\`E,\k E,\v E},
3902   e = {\`e,\`e,\^e,\`e,\k e,\v e},
3903   f = {027}, % ff
3904   G = {\u G},
3905   q = {\u q},

```

```

3906 I = {\`I,\`I,\^I,\"I,.I},
3907 i = {\`i,\`i,\^i,\"i,i},
3908 j = {\j},
3909 L = {\L,\`L,\v L},
3910 l = {\l,\`l,\v l},
3911 N = {\`N,\~N,\v N},
3912 n = {\`n,\~n,\v n},
3913 O = {\0,\`0,\`0,\^0,\~0,\"0,\H 0},
3914 o = {\0,\`o,\`o,\^o,\~o,\"o,\H o},
3915 R = {\`R,\v R},
3916 r = {\`r,\v r},
3917 S = {\`S,\c S,\v S,\SS},
3918 s = {\`s,\c s,\v s},
3919 T = {\c T,\v T},
3920 t = {\c t,\v t},
3921 U = {\`U,\`U,\^U,\"U,\H U,\r U},
3922 u = {\`u,\`u,\^u,\"u,\H u,\r u},
3923 Y = {\`Y,\"Y},
3924 y = {\`y,\"y},
3925 Z = {\`Z,.Z,\v Z},
3926 z = {\`z,.z,\v z}

```

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

```

3927 % - = {127},
3928 }
3929

```

### 15.5.3 LY1

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

```

3930 \DeclareCharacterInheritance
3931 { encoding = LY1 }
3932 { A = {\`A,\`A,\^A,\~A,\"A,\r A},
3933   a = {\`a,\`a,\^a,\~a,\"a,\r a},
3934   C = {\c C},
3935   c = {\c c},
3936   D = {\DH},
3937   E = {\`E,\`E,\^E,\"E},
3938   e = {\`e,\`e,\^e,\"e},
3939   f = {011}, % ff
3940   I = {\`I,\`I,\^I,\"I},
3941   i = {\`i,\`i,\^i,\"i,i},
3942   L = {\L},
3943   l = {\l},
3944   N = {\~N},
3945   n = {\~n},
3946   O = {\`0,\`0,\^0,\~0,\"0,\0},
3947   o = {\`o,\`o,\^o,\~o,\"o,\o},
3948   S = {\v S},
3949   s = {\v s},
3950   U = {\`U,\`U,\^U,\"U},
3951   u = {\`u,\`u,\^u,\"u},
3952   Y = {\`Y,\"Y},
3953   y = {\`y,\"y},
3954   Z = {\v Z},
3955   z = {\v z}
3956 }
3957

```

### 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.

```

3958 \DeclareCharacterInheritance
3959 { encoding = OT4 }
3960 { A = {\k A},
3961   a = {\k a},
3962   C = {\'C},
3963   c = {\'c},
3964   E = {\k E},
3965   e = {\k e},
3966   f = {011}, % ff
3967   i = {\i},
3968   j = {\j},
3969   L = {\L},
3970   l = {\l},
3971   N = {\'N},
3972   n = {\'n},
3973   O = {\0,\'0},
3974   o = {\o,\'o},
3975   S = {\'S},
3976   s = {\'s},
3977   Z = {\'Z,\.Z},
3978   z = {\'z,\.z}
3979 }
3980

```

### 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.

```

3981 \DeclareCharacterInheritance
3982 { encoding = QX }
3983 { A = {\^A,\'A,\^A,\-A,\"A,\k A,\AA},
3984   a = {\^a,\'a,\^a,\-a,\"a,\k a,\aa},
3985   C = {\'C,\c C},
3986   c = {\'c,\c c},
3987   D = {\DH},
3988   E = {\^E,\'E,\^E,\"E,\k E},
3989   e = {\^e,\'e,\^e,\"e,\k e},
3990   f = {011}, % ff
3991   I = {\^I,\'I,\^I,\"I,\k I},
3992   i = {\^i,\'i,\^i,\"i,\k i,\i},
3993   j = {\j},
3994   L = {\L},
3995   l = {\l},
3996   N = {\'N,\-N},
3997   n = {\'n,\-n},
3998   O = {\0,\'0,\^0,\-0,\"0},
3999   o = {\o,\'o,\^o,\-o,\"o},
4000   S = {\'S,\c S,\v S},
4001   s = {\'s,\c s,\v s},
4002   T = {\c T},
4003   t = {\c t},
4004   U = {\^U,\'U,\^U,\"U,\k U},
4005   u = {\^u,\'u,\^u,\"u,\k u},
4006   Y = {\'Y,\"Y},

```

<sup>13</sup> Thanks to *Maciej Eder*.

```

4007     y = {\'y,\"y},
4008     Z = {\'Z,\"Z,\"v Z},
4009     z = {\'z,\"z,\"v z},
4010     . = \textellipsis
4011 }
4012

```

### 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.

```

4013 \DeclareCharacterInheritance
4014 { encoding = T5 }
4015 { A = {\'A,\"A,\"-A,\"h A,\"d A,\"^A,\"u A,
4016       \"\Acircumflex,\"'\Acircumflex,\"-\Acircumflex,\"h\Acircumflex,\"d\Acircumflex,
4017       \"\Abreve,\"'\Abreve,\"-\Abreve,\"h\Abreve,\"d\Abreve},
4018     a = {\'a,\"a,\"-a,\"h a,\"d a,\"^a,\"u a,
4019       \"\acircumflex,\"'\acircumflex,\"-\acircumflex,\"h\acircumflex,\"d\acircumflex,
4020       \"\abreve,\"'\abreve,\"-\abreve,\"h\abreve,\"d\abreve},
4021     D = {\DJ},
4022     d = {\dj},
4023     E = {\'E,\"E,\"-E,\"h E,\"d E,\"^E,
4024       \"\Ecircumflex,\"'\Ecircumflex,\"-\Ecircumflex,\"h\Ecircumflex,\"d\Ecircumflex},
4025     e = {\'e,\"e,\"-e,\"h e,\"d e,\"^e,
4026       \"\ecircumflex,\"'\ecircumflex,\"-\ecircumflex,\"h\ecircumflex,\"d\ecircumflex},
4027     I = {\'I,\"I,\"-I,\"h I,\"d I},
4028     i = {\'i,\"i,\"-i,\"h i,\"d i,\"i},
4029     O = {\'O,\"O,\"-O,\"h O,\"d O,\"^O,\"horn O,
4030       \"\Ocircumflex,\"'\Ocircumflex,\"-\Ocircumflex,\"h\Ocircumflex,\"d\Ocircumflex,
4031       \"\Ohorn,\"'\Ohorn,\"-\Ohorn,\"h\Ohorn,\"d\Ohorn},
4032     o = {\'o,\"o,\"-o,\"h o,\"d o,\"^o,\"horn o,
4033       \"\ocircumflex,\"'\ocircumflex,\"-\ocircumflex,\"h\ocircumflex,\"d\ocircumflex,
4034       \"\ohorn,\"'\ohorn,\"-\ohorn,\"h\ohorn,\"d\ohorn},
4035     U = {\'U,\"U,\"-U,\"h U,\"d U,\"horn U,
4036       \"\Uhorn,\"'\Uhorn,\"-\Uhorn,\"h\Uhorn,\"d\Uhorn},
4037     u = {\'u,\"u,\"-u,\"h u,\"d u,\"horn u,
4038       \"\uhorn,\"'\uhorn,\"-\uhorn,\"h\uhorn,\"d\uhorn},
4039     Y = {\'Y,\"Y,\"-Y,\"h Y,\"d Y},
4040     y = {\'y,\"y,\"-y,\"h y,\"d y}
4041 }
4042
4043 </m-t>

```

### 15.5.7 Euro symbols

Make Euro symbols settings simpler.

```

4044 < *zpeu>
4045 \DeclareCharacterInheritance
4046 { encoding = U,
4047   family   = {zpeu,zpeus,eurosans} }
4048 { E = 128 }
4049
4050 </zpeu>
4051 < *mvs>
4052 \DeclareCharacterInheritance
4053 { encoding = OT1,
4054   family   = mvs }
4055 { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
4056

```

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.

```
4057 \DeclareCharacterInheritance
4058   { encoding = U,
4059     family   = mvs }
4060   { 164 = {099,100,101} }
4061
4062 /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.

```
4063 <*m-t>
4064 %%% -----
4065 %%% TRACKING/LETTERSPACING
4066
4067 \SetTracking
4068   [ no ligatures = {f} ]
4069   { encoding      = {OT1,T1,LY1,OT4,QX} }
4070   {}
4071
```

## 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).

```
4072 %%% -----
4073 %%% EXPANSION
4074
4075 \SetExpansion
4076   [ name      = default      ]
4077   { encoding = {OT1,OT4,QX,T1,LY1} }
4078   {
4079     A = 500,    a = 700,
4080     \AE = 500,  \ae = 700,
4081     B = 700,    b = 700,
4082     C = 700,    c = 700,
4083     D = 500,    d = 700,
4084     E = 700,    e = 700,
4085     F = 700,
4086     G = 500,    g = 700,
4087     H = 700,    h = 700,
4088     K = 700,    k = 700,
4089     M = 700,    m = 700,
4090     N = 700,    n = 700,
4091     O = 500,    o = 700,
4092     \OE = 500,  \oe = 700,
4093     P = 700,    p = 700,
4094     Q = 500,    q = 700,
4095     R = 700,
4096     S = 700,    s = 700,
4097     U = 700,    u = 700,
4098     W = 700,    w = 700,
4099     Z = 700,    z = 700,
4100     2 = 700,
4101     3 = 700,
```

```

4102     6 = 700,
4103     8 = 700,
4104     9 = 700
4105   }
4106

```

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

```

4107 \SetExpansion
4108   [ name      = T5 ]
4109   { encoding = T5 }
4110   {
4111     A = 500,      a = 700,
4112     B = 700,      b = 700,
4113     C = 700,      c = 700,
4114     D = 500,      d = 700,
4115     E = 700,      e = 700,
4116     F = 700,
4117     G = 500,      g = 700,
4118     H = 700,      h = 700,
4119     K = 700,      k = 700,
4120     M = 700,      m = 700,
4121     N = 700,      n = 700,
4122     O = 500,      o = 700,
4123     P = 700,      p = 700,
4124     Q = 500,      q = 700,
4125     R = 700,
4126     S = 700,      s = 700,
4127     U = 700,      u = 700,
4128     W = 700,      w = 700,
4129     Z = 700,      z = 700,
4130     2 = 700,
4131     3 = 700,
4132     6 = 700,
4133     8 = 700,
4134     9 = 700
4135   }
4136
4137 </m-t>

```

## 15.8 Character Protrusion

```

4138 %%% -----
4139 %%% PROTRUSION
4140

```

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.

```

4141 <*cfg-t>
4142 \SetProtrusion
4143 <m-t> [ name = default ]

```

We also create configuration files for the fonts Bitstream Charter (NFSS code bch),

```

4144 <bch> [ name = bch-default ]

```

Computer Modern Roman (cmr),

```

4145 <cmr> [ name = cmr-default ]

```

Adobe Garamond (pad, padx, padj),

```

4146 <pad> [ name = pad-default ]

```

Minion<sup>14</sup> (pmnx, pmnj),

```

4147 <pmn> [ name = pmnj-default ]

```

Palatino (ppl, pplx, pplj),

```

4148 <ppl> [ name = ppl-default ]

```

Times (ptm, ptmx, ptmj),

```

4149 <ptm> [ name = ptm-default ]

```

URW Garamond (ugm)

```

4150 <ugm> [ name = ugm-default ]

```

and Bitstream Letter Gothic (blg).

```

4151 <blg> [ name = blg-default ]
4152 <m-t> { encoding = OT1 }
4153 <cmr> { }
4154 <bch|blg|pad|pmn|ugm> { encoding = OT1,
4155 <ppl|ptm> { encoding = {OT1,OT4},
4156 <bch> family = bch }
4157 <blg> family = blg }
4158 <pad> family = {pad,padx,padj} }
4159 <pmn> family = pmnj }
4160 <ppl> family = {ppl,pplx,pplj} }
4161 <ptm> family = {ptm,ptmx,ptmj} }
4162 <ugm> family = ugm }
4163 {
4164 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> A = {50,50},
4165 <ugm> A = {50,100},

```

14 Contributed by Harald Harders ([h.harders@tu-bs.de](mailto:h.harders@tu-bs.de))

```

4166 <m-t|pad|ptm> \AE = {50, },
4167 <ugm> \AE = {150,50},
4168 <ugm> B = { ,50},
4169 <bch|pad|pmn|ugm> C = {50, },
4170 <bch|pad|pmn> D = { ,50},
4171 <ugm> D = { ,70},
4172 <ugm> E = { ,50},
4173 <m-t|bch|cmr|pad|pmn|ptm> F = { ,50},
4174 <ugm> F = { ,70},
4175 <bch|pad|pmn> G = {50, },
4176 <ugm> G = {50,50},
4177 <blg> I = {150,150},
4178 <m-t|cmr|pad|pmn|ppl|ptm|ugm> J = {50, },
4179 <bch|blg> J = {100, },
4180 <!blg> K = { ,50},
4181 <blg> K = {50, },
4182 <m-t|bch|cmr|pad|pmn|ppl> L = { ,50},
4183 <blg> L = { ,150},
4184 <ptm> L = { ,80},
4185 <ugm> L = { ,120},
4186 <bch|pad|pmn|ugm> O = {50,50},
4187 <pad|pmn> \OE = {50, },
4188 <ugm> \OE = {50,50},
4189 <blg> P = { ,100},
4190 <ugm> P = { ,50},
4191 <bch|pad|pmn> Q = {50,70},
4192 <ugm> Q = {50,50},
4193 <bch> R = { ,50},
4194 <ugm> R = { ,70},
4195 <m-t|bch|cmr|pad|pmn|ppl|ptm> T = {50,50},
4196 <blg> T = {100,100},
4197 <ugm> T = {70,70},
4198 <m-t|bch|cmr|pad|pmn|ppl|ptm> V = {50,50},
4199 <blg|ugm> V = {70,70},
4200 <m-t|bch|cmr|pad|pmn|ppl|ptm> W = {50,50},
4201 <ugm> W = {70,70},
4202 <m-t|bch|cmr|pad|pmn|ppl|ptm> X = {50,50},
4203 <ugm> X = {50,70},
4204 <m-t|bch|cmr|pad|pmn|ppl> Y = {50,50},
4205 <blg|ptm|ugm> Y = {80,80},
4206 <ugm> Z = {50,50},
4207 <blg> f = {150,100},
4208 <blg> i = {150,150},
4209 <blg> j = {100,100},
4210 <m-t|bch|cmr|pad|pmn|ppl|ptm> k = { ,50},
4211 <ugm> k = { ,70},
4212 <blg> l = {150,150},
4213 <pmn> l = { , -50},
4214 <pad|ppl> p = {50,50},
4215 <ugm> p = { ,50},
4216 <pad|ppl> q = {50, },
4217 <!blg> r = { ,50},
4218 <blg> r = {100, 80},
4219 <cmr|pad|pmn> t = { ,70},
4220 <bch> t = { ,50},
4221 <blg> t = {150, 80},
4222 <ugm> t = { ,100},
4223 <m-t|bch|cmr|pad|pmn|ppl|ptm> v = {50,50},
4224 <blg> v = {100,100},
4225 <ugm> v = {50,70},
4226 <m-t|bch|cmr|pad|pmn|ppl|ptm> w = {50,50},
4227 <ugm> w = {50,70},
4228 <!blg> x = {50,50},

```

```

4229 <blg>      x = {100,100},
4230 <m-t|bch|pad|pmn>      y = { ,50},
4231 <blg>      y = { 50,100},
4232 <cmr|ppl|ptm>      y = {50,70},
4233 <ugm>      y = { ,70},

4234 <cmr>      0 = { ,50},
4235 <m-t>      1 = {50,50},
4236 <bch|blg|pad|ptm|ugm>      1 = {150,150},
4237 <cmr>      1 = {100,200},
4238 <pmn>      1 = { ,50},
4239 <ppl>      1 = {100,100},
4240 <bch|cmr|pad|ugm>      2 = {50,50},
4241 <blg>      2 = { ,100},
4242 <bch|pmn>      3 = {50, },
4243 <cmr|pad|ugm>      3 = {50,50},
4244 <blg>      3 = {100, },
4245 <m-t|pad>      4 = {50,50},
4246 <bch>      4 = {100,50},
4247 <blg>      4 = {100, },
4248 <cmr|ugm>      4 = {70,70},
4249 <pmn>      4 = {50, },
4250 <ptm>      4 = {70, },
4251 <cmr>      5 = { ,50},
4252 <pad>      5 = {50,50},
4253 <bch>      6 = {50, },
4254 <cmr>      6 = { ,50},
4255 <pad>      6 = {50,50},
4256 <m-t>      7 = {50,50},
4257 <bch|pad|pmn|ugm>      7 = {50,80},
4258 <blg>      7 = {100,100},
4259 <cmr|ptm>      7 = {50,100},
4260 <ppl>      7 = { ,50},
4261 <cmr>      8 = { ,50},
4262 <bch|pad>      9 = {50,50},
4263 <cmr>      9 = { ,50},
4264 <m-t|cmr|pad|pmn|ppl|ptm|ugm>      . = { ,700},
4265 <bch>      . = { ,600},
4266 <blg>      . = {400,500},
4267 <!blg>      {,}= { ,500},
4268 <blg>      {,}= {300,400},
4269 <m-t|cmr|pad|pmn|ppl|ptm|ugm>      : = { ,500},
4270 <bch>      : = { ,400},
4271 <blg>      : = {300,400},
4272 <m-t|bch|pad|pmn|ptm>      ; = { ,300},
4273 <blg>      ; = {200,300},
4274 <cmr|ppl>      ; = { ,500},
4275 <ugm>      ; = { ,400},
4276 <!blg>      ! = { ,100},
4277 <blg>      ! = {200,200},
4278 <m-t|pad|pmn|ptm>      ? = { ,100},
4279 <bch|cmr|ppl|ugm>      ? = { ,200},
4280 <blg>      ? = {150,150},
4281 <pmn>      " = {300,300},
4282 <m-t|bch|cmr|pad|pmn|ppl>      @ = {50,50},
4283 <ptm>      @ = {100,100},
4284 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm>      ~ = {200,250},
4285 <ugm>      ~ = {300,350},
4286 <pad|ppl|ptm>      & = {50,100},
4287 <ugm>      & = { ,100},
4288 <m-t|cmr|pad|pmn>      \% = {50,50},
4289 <bch>      \% = { ,50},
4290 <ppl|ptm>      \% = {100,100},

```

```

4291 <ugm> \% = {50,100},
4292 <blg> \# = {100,100},
4293 <m-t|ppl|ptm|ugm> * = {200,200},
4294 <bch|pmn> * = {200,300},
4295 <blg> * = {150,200},
4296 <cmr|pad> * = {300,300},
4297 <m-t|cmr|ppl|ptm> + = {250,250},
4298 <bch> + = {150,250},
4299 <pad> + = {300,300},
4300 <blg|pmn> + = {150,200},
4301 <ugm> + = {250,300},
4302 <blg|ugm> {=} = {200,200},
4303 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
4304 <bch|ugm> ( = {200, }, ) = { ,200},
4305 <blg> ( = {300, }, ) = { ,300},
4306 <cmr|ppl> ( = {100, }, ) = { ,300},
4307 <bch|pmn> [ = {100, }, ] = { ,100},
4308 <blg> [ = {300,100}, ] = { ,300},

4309 <m-t|pad|pmn|ptm> / = {100,200},
4310 <bch> / = { ,200},
4311 <blg> / = {300,300},
4312 <cmr|ppl> / = {200,300},
4313 <ugm> / = {100,300},
4314 <m-t|ptm> - = {500,500},
4315 <bch|cmr|ppl> - = {400,500},
4316 <blg> - = {300,400},
4317 <pad> - = {300,500},
4318 <pmn> - = {200,400},
4319 <ugm> - = {500,600},
4320 <blg> <= {200,100}, >= {100,200},
4321 <blg> _ = {150,250},
4322 <blg> | = {250,250},
4323 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
4324 <bch> \textendash = {200,300}, \textendash = {150,250},
4325 <cmr> \textendash = {400,300}, \textendash = {300,200},
4326 <pad|ppl|ptm> \textendash = {300,300}, \textendash = {200,200},
4327 <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).

```

4328 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
4329 <blg> \textquoteleft = {400,600}, \textquoteright = {400,600},
4330 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
4331 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
4332 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
4333 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
4334 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300}
4335 <blg> \textquotedblright = {300,400}
4336 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600}
4337 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4338 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400}
4339 }
4340

```

Greek uppercase letters are in OT1 encoding only.

```

4341 <*cmr>
4342 \SetProtrusion
4343 [ name = cmr-OT1,
4344   load = cmr-default ]
4345 { encoding = {OT1,OT4},
4346   family = cmr }

```

```

4347 {
4348   \AE = { 50, },
4349   "00 = { ,150}, % \Gamma
4350   "01 = {100,100}, % \Delta
4351   "02 = { 50, 50}, % \Theta
4352   "03 = {100,100}, % \Lambda
4353   "06 = { 50, 50}, % \Sigma
4354   "07 = {100,100}, % \Upsilon
4355   "08 = { 50, 50}, % \Phi
4356   "09 = { 50, 50} % \Psi

```

Remaining slots can be found in the source file.

```

4357 }
4358
4359 </cmr>

```

T1 and LY1 encodings contain some more characters. The default list will be loaded first.

```

4360 \SetProtrusion
4361 <m-t> [ name = T1-default,
4362 <bch> [ name = bch-T1,
4363 <blg> [ name = blg-T1,
4364 <cmr> [ name = cmr-T1,
4365 <pad> [ name = pad-T1,
4366 <pmn> [ name = pmnj-T1,
4367 <ppl> [ name = ppl-T1,
4368 <ptm> [ name = ptm-T1,
4369 <ugm> [ name = ugm-T1,
4370 <m-t> load = default ]
4371 <bch> load = bch-default ]
4372 <blg> load = blg-default ]
4373 <cmr> load = cmr-default ]
4374 <pad> load = pad-default ]
4375 <pmn> load = pmnj-default ]
4376 <ppl> load = ppl-default ]
4377 <ptm> load = ptm-default ]
4378 <ugm> load = ugm-default ]
4379 <m-t> { encoding = {T1,LY1} }
4380 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4381 <blg|ptm|ugm> { encoding = {T1},
4382 <bch> family = bch }
4383 <blg> family = blg }
4384 <cmr> family = cmr }
4385 <pad> family = {pad,padx,padj} }
4386 <pmn> family = pmnj }
4387 <ppl> family = {ppl,pplx,pplj} }
4388 <ptm> family = {ptm,ptmx,ptmj} }
4389 <ugm> family = ugm }
4390 {
4391 <cmr> \AE = {50, },
4392 <bch> \OE = {50, },
4393 <pmn> \TH = { ,50},
4394 <blg> \v L = { ,250},
4395 <blg> \v d = { ,250},
4396 <blg> \v l = { ,250},
4397 <blg> \v t = { ,250},
4398 <blg> 127 = {300,400},
4399 <blg> 156 = {100, }, % IJ
4400 <blg> 188 = { 80, 80}, % ij
4401 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
4402 <cmr> _ = {200,200},
4403 <ugm> _ = {100,200},

```

```

4404 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
4405 <bch> \textbackslash = {150,200},
4406 <blg> \textbackslash = {250,300},
4407 <cmr|ppl> \textbackslash = {200,300},
4408 <ugm> \textbackslash = {100,300},
4409 <ugm> \textbar = {200,200},
4410 <blg> \textendash = {300,300}, \textemdash = {150,150},
4411 <blg> \textquotedbl = {300,400}, \textquotedblleft = {300,400},
4412 <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.

```

4413 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4414 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
4415 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4416 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4417 <blg> \guilsinglleft = {300,500}, \guilsinglright = {300,500},
4418 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4419 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4420 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
4421 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
4422 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
4423 <blg|pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
4424 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4425 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
4426 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
4427 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4428 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
4429 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
4430 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
4431 <pmn> \textless = {100, }, \textgreater = { ,100},
4432 <pmn> \textvisiblespace = {100,100} % not in LY1
4433 }
4434

```

The lmodern fonts, on the other hand, restore the original kerning from the OT1 fonts, and so do we. Silly, isn't it?

```

4435 <*cmr>
4436 \SetProtrusion
4437 [ name = lmr-T1,
4438   load = cmr-T1 ]
4439 { encoding = {T1,LY1},
4440   family = lmr }
4441 {
4442   \textquotedblleft = {500,300},
4443   \quotedblbase = {500,300}
4444 }
4445
4446 </cmr>

```

Settings for the QX encoding (generic and Times). It also includes some glyphs otherwise in TS1.

```

4447 <*m-t|ptm>
4448 \SetProtrusion
4449 <m-t> [ name = QX-default,
4450 <ptm> [ name = ptm-QX,
4451 <m-t>   load = default ]
4452 <ptm>   load = ptm-default ]
4453 <m-t> { encoding = QX }
4454 <ptm> { encoding = QX,
4455 <ptm>   family = {ptm,ptmx,ptmj} }

```

```

4456 {
4457   <ptm>      * = {200,200},
4458   {=} = {100,100},
4459   \textunderscore = {100,100},
4460   \textbackslash = {100,200},
4461   \quotedblbase = {400,400},
4462   <m-t>      \guillemotleft = {200,200}, \guillemotright = {200,200},
4463   <ptm>      \guillemotleft = {300,300}, \guillemotright = {200,400},
4464   \textexclamdown = {100, }, \textquestiondown = {100, },
4465   <m-t>      \textbraceleft = {400,200}, \textbraceright = {200,400},
4466   <ptm>      \textbraceleft = {200,200}, \textbraceright = {200,300},
4467   \textless = {200,100}, \textgreater = {100,200},
4468   \textminus = {200,200}, \textdegree = {300,300},
4469   <m-t>      \copyright = {100,100}, \textregistered = {100,100},
4470   <ptm>      \copyright = {100,150}, \textregistered = {100,150},
4471   <ptm>      \textxgeq = { ,100}, \textxleq = {100, },
4472   <ptm>      \textalpha = { , 50}, \textDelta = { 70, 70},
4473   <ptm>      \textpi = { 50, 80}, \textSigma = { , 70},
4474   <ptm>      \textmu = { , 80}, \texteuro = { 50, 50},
4475   <ptm>      \textellipsis = {150,200}, \textasciitilde = { 80, 80},
4476   <ptm>      \textapprox = { 50, 50}, \textinfty = {100,100},
4477   <ptm>      \textdagger = {150,150}, \textdaggerdbl = {100,100},
4478   <ptm>      \textdiv = { 50,150}, \textsection = { 80, 80},
4479   <ptm>      \texttimes = {100,150}, \textpm = { 50, 80},
4480   <ptm>      \textbullet = {150,150}, \textperiodcentered = {300,300},
4481   <ptm>      \textquotesingle = {500,500}, \textquotedbl = {300,300},
4482   <ptm>      \textperthousand = { ,50}
4483 }
4484
4485 </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.

```

4486 <*cmr|bch>
4487 \SetProtrusion
4488 <cmr> [ name = cmr-T5,
4489 <cmr>   load = cmr-default ]
4490 <bch> [ name = bch-T5,
4491 <bch>   load = bch-default ]
4492 { encoding = T5,
4493 <cmr>   family = cmr }
4494 <bch>   family = bch }
4495 {
4496 <bch>   _ = {100,100},
4497 <bch>   \textbackslash = {150,200},
4498 <cmr>   \textbackslash = {200,300},
4499 <cmr>   \textquotedblleft = {200,600},
4500 <cmr>   \textquotedbl = {300,300},
4501 <bch>   \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4502 <cmr>   \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4503 <bch>   \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4504 <cmr>   \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4505 <bch>   \guillemotleft = {200,200}, \guillemotright = {150,300},
4506 <cmr>   \guillemotleft = {300,200}, \guillemotright = {100,400},
4507 <bch>   \textbraceleft = {200, }, \textbraceright = { ,300},
4508 <cmr>   \textbraceleft = {400,200}, \textbraceright = {200,400},
4509   \textless = {200,100}, \textgreater = {100,200}
4510 }
4511
4512 </cmr|bch>
4513 <*pmn>
4514 \SetProtrusion

```

```

4515 [ name      = pmnx-OT1,
4516     load      = pmnj-default ]
4517 { encoding = OT1,
4518     family   = pmnx }
4519 {
4520     1 = {230,180}
4521 }
4522
4523 \SetProtrusion
4524 [ name      = pmnx-T1,
4525     load      = pmnj-T1 ]
4526 { encoding = {T1,LY1},
4527     family   = pmnx }
4528 {
4529     1 = {230,180}
4530 }
4531
4532 </pmn>

```

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

```

4533 <*ptm>
4534 \SetProtrusion
4535 [ name      = ptm-LY1,
4536     load      = ptm-T1 ]
4537 { encoding = LY1,
4538     family   = {ptm,ptmx,ptmj} }
4539 {
4540     -                      = {100,100},
4541     \texttrademark         = {100,100},
4542     \textregistered        = {100,100},
4543     \textcopyright         = {100,100},
4544     \textdegree            = {300,300},
4545     \textminus             = {200,200},
4546     \textellipsis          = {150,200},
4547     \texteuro              = {  ,  }, % ?
4548     \textcent              = {100,100},
4549     \textquotesingle       = {500,500},
4550     \textflorin            = { 50, 70},
4551     \textdagger            = {150,150},
4552     \textdaggerdbl         = {100,100},
4553     \textperthousand       = {  , 50},
4554     \textbullet            = {150,150},
4555     \textonesuperior       = {100,100},
4556     \texttwosuperior       = { 50, 50},
4557     \textthreesuperior     = { 50, 50},
4558     \textperiodcentered    = {300,300},
4559     \textplusminus         = { 50, 80},
4560     \textmultiply          = {100,100},
4561     \textdivide            = { 50,150}

```

Remaining slots in the source file.

```

4562 }
4563
4564 </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.

```

4565 \SetProtrusion
4566 <m-t> [ name      = OT1-it  ]
4567 <bch> [ name      = bch-it  ]
4568 <blg> [ name      = blg-it,
4569 <blg> [ load      = blg-default ]
4570 <cmr> [ name      = cmr-it  ]
4571 <pad> [ name      = pad-it  ]
4572 <pmn> [ name      = pmnj-it  ]
4573 <ppl> [ name      = ppl-it  ]
4574 <ptm> [ name      = ptm-it  ]
4575 <ugm> [ name      = ugm-it  ]
4576 <m-t|bch|blg|pad|pmn|ugm> { encoding = OT1,
4577 <ppl|ptm> { encoding = {OT1,OT4},
4578 <bch>      family   = bch,
4579 <blg>      family   = blg,
4580 <pad>      family   = {pad,padx,padj},
4581 <pmn>      family   = pmnj,
4582 <ppl>      family   = {ppl,pplx,pplj},
4583 <ptm>      family   = {ptm,ptmx,ptmj},
4584 <ugm>      family   = ugm,
4585 <m-t|bch|pad|pmn|ppl|ptm> shape   = {it,sl} }
4586 <blg|ugm> shape   = it }
4587 <cmr> { }
4588 {
4589 <cmr|ptm> A = {100,50},
4590 <pad|pmn> A = {50, },
4591 <ugm>     A = { ,150},
4592 <ppl>     A = {50,50},
4593 <ptm>     \AE = {100, },
4594 <pad|ppl> \AE = {50, },
4595 <pmn>     \AE = { , -50},
4596 <cmr|pad|ppl|ptm> B = {50, },
4597 <pmn>     B = {20,-50},
4598 <bch|ppl|ptm|ugm> C = {50, },
4599 <cmr|pad> C = {100, },
4600 <pmn>     C = {50,-50},
4601 <cmr|pad|ppl|ptm> D = {50,50},
4602 <pmn>     D = {20, },
4603 <cmr|pad|ppl|ptm> E = {50, },
4604 <pmn>     E = {20,-50},
4605 <cmr|pad|ptm>    F = {100, },
4606 <pmn>          F = {10, },
4607 <ppl>          F = {50, },
4608 <bch|ppl|ptm|ugm> G = {50, },
4609 <cmr|pad>      G = {100, },
4610 <pmn>          G = {50,-50},
4611 <cmr|pad|ppl|ptm> H = {50, },
4612 <cmr|pad|ptm>   I = {50, },
4613 <pmn>          I = {20,-50},
4614 <cmr|ptm>      J = {100, },
4615 <pad>          J = {50, },
4616 <pmn>          J = {20, },
4617 <cmr|pad|ppl|ptm> K = {50, },
4618 <pmn>          K = {20, },
4619 <cmr|pad|ppl|ptm> L = {50, },
4620 <pmn>          L = {20,50},
4621 <ugm>          L = { ,100},
4622 <cmr|ptm>      M = {50, },
4623 <pmn>          M = { , -30},
4624 <cmr|ptm>      N = {50, },
4625 <pmn>          N = { , -30},

```

```

4626 <bch|pmn|ppl|ptm>    0 = {50, },
4627 <cmr|pad>            0 = {100, },
4628 <ugm>                0 = {70,50},
4629 <pmn|ppl|ptm>        \OE = {50, },
4630 <pad>                \OE = {100, },
4631 <cmr|pad|ppl|ptm>    P = {50, },
4632 <pmn>                P = {20,-50},
4633 <bch|pmn|ppl|ptm>    Q = {50, },
4634 <cmr|pad>            Q = {100, },
4635 <ugm>                Q = {70,50},
4636 <cmr|pad|ppl|ptm>    R = {50, },
4637 <pmn>                R = {20, },
4638 <bch|cmr|pad|ppl|ptm> S = {50, },
4639 <pmn>                S = {20,-30},
4640 <bch|cmr|pad|ppl|ptm> $ = {50, },
4641 <pmn>                $ = {20,-30},
4642 <bch|pmn|ugm>        T = {70, },
4643 <cmr|pad|ppl|ptm>    T = {100, },
4644 <cmr|pad|ppl|ptm>    U = {50, },
4645 <pmn>                U = {50,-50},
4646 <cmr|pad|pmn|ugm>    V = {100, },
4647 <ppl|ptm>            V = {100,50},
4648 <cmr|pad|pmn|ugm>    W = {100, },
4649 <ppl>                W = {50, },
4650 <ptm>                W = {100,50},
4651 <cmr|ppl|ptm>        X = {50, },
4652 <cmr|ptm>            Y = {100, },
4653 <pmn>                Y = {50, },
4654 <ppl>                Y = {100,50},
4655 <pmn>                Z = { , -50},
4656 <pmn>                d = { , -50},
4657 <pad|pmn>            f = { , -100},
4658 <pmn>                i = { , -30},
4659 <pmn>                j = { , -30},
4660 <pmn>                l = { , -100},
4661 <bch>                o = {50,50},
4662 <bch>                p = { , 50},
4663 <pmn>                p = {-50, },
4664 <bch>                q = {50, },
4665 <pmn>                r = { , 50},
4666 <bch>                t = { , 50},
4667 <pmn|ugm>            v = {50, },
4668 <bch>                w = { , 50},
4669 <pmn|ugm>            w = {50, },
4670 <bch>                y = { , 50},
4671 <cmr>                0 = {100, },
4672 <bch|ptm>            1 = {150,100},
4673 <cmr>                1 = {200,50},
4674 <pad>                1 = {150, },
4675 <pmn>                1 = {50, },
4676 <ppl>                1 = {100, },
4677 <ugm>                1 = {150,150},
4678 <cmr>                2 = {100,-100},
4679 <pad|ppl|ptm>        2 = {50, },
4680 <pmn>                2 = {-50, },
4681 <bch>                3 = {50, },
4682 <cmr>                3 = {100,-100},
4683 <pmn>                3 = {-100, },
4684 <ptm>                3 = {100,50},
4685 <bch>                4 = {100, },
4686 <cmr|pad>            4 = {150, },
4687 <ppl|ptm>            4 = {50, },
4688 <cmr>                5 = {100, },

```

```

4689 <ptm>      5 = {50, },
4690 <bch>      6 = {50, },
4691 <cmr>      6 = {100, },
4692 <bch|pad|ptm> 7 = {100, },
4693 <cmr>      7 = {200,-150},
4694 <pmn>      7 = {20, },
4695 <ppl>      7 = {50, },
4696 <cmr>      8 = {50,-50},
4697 <cmr>      9 = {100,-100},
4698 <m-t|cmr|pad|pmn|ppl> . = { ,500},
4699 <blg>      . = {400,600},
4700 <bch|ptm|ugm> . = { ,700},
4701 <blg>      {,}= {300,500},
4702 <m-t|cmr|pad|pmn|ppl> {,}= { ,500},
4703 <bch|ugm>    {,}= { ,600},
4704 <ptm>      {,}= { ,700},
4705 <m-t|cmr|pad|ppl> : = { ,300},
4706 <bch|ugm>    : = { ,400},
4707 <pmn>      : = { ,200},
4708 <ptm>      : = { ,500},
4709 <m-t|cmr|pad|ppl> ; = { ,300},
4710 <bch|ugm>    ; = { ,400},
4711 <pmn>      ; = { ,200},
4712 <ptm>      ; = { ,500},
4713 <ptm>      != { ,100},
4714 <bch>      ? = { ,200},
4715 <ptm>      ? = { ,100},
4716 <ppl>      ? = { ,300},
4717 <pmn>      " = {400,200},
4718 <m-t|pad|pmn|ppl|ptm> & = {50,50},
4719 <bch>      & = { ,80},
4720 <cmr>      & = {100,50},
4721 <ugm>      & = {50,100},
4722 <m-t|cmr|pad|pmn> \% = {100, },
4723 <bch>      \% = {50,50},
4724 <ppl|ptm>    \% = {100,100},
4725 <ugm>      \% = {100,50},
4726 <m-t|pmn|ppl> * = {200,200},
4727 <bch>      * = {300,200},
4728 <cmr>      * = {400,100},
4729 <pad>      * = {500,100},
4730 <ptm|ugm>    * = {400,200},
4731 <m-t|cmr|pmn|ppl> + = {150,200},
4732 <bch|ugm>    + = {250,250},
4733 <pad|ptm>    + = {250,200},
4734 <m-t|pad|pmn|ppl> @ = {50,50},
4735 <bch>      @ = {80,50},
4736 <cmr>      @ = {200,50},
4737 <ptm>      @ = {150,150},
4738 <m-t|bch|ugm> ~ = {150,150},
4739 <cmr|pad|pmn|ppl|ptm> - = {200,150},
4740 <ugm>      {=} = {200,200},
4741 <!blg>      ( = {200, }, ) = { ,200},
4742 <m-t|cmr|pad|ppl|ptm|ugm> / = {100,200},
4743 <bch>      / = { ,150},
4744 <pmn>      / = {100,150},
4745 <m-t>      - = {300,300},
4746 <bch|pad>   - = {300,400},
4747 <pmn>      - = {200,300},
4748 <cmr>      - = {500,300},
4749 <ppl>      - = {300,500},
4750 <ptm>      - = {500,500},
4751 <ugm>      - = {400,700},

```

```

4752 <blg>      _ = {0,300},
4753 <m-t|pmn>   \textendash      = {200,200}, \textemdash      = {150,150},
4754 <bch>       \textendash      = {200,300}, \textemdash      = {150,200},
4755 <cmr>       \textendash      = {500,300}, \textemdash      = {400,200},
4756 <pad|ppl|ptm|ugm> \textendash      = {300,300}, \textemdash      = {200,200},
4757 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
4758 <blg>       \textquoteleft = {400,400}, \textquoteright = {400,400},
4759 <cmr|pad>    \textquoteleft = {800,200}, \textquoteright = {800,200},
4760 <ppl>        \textquoteleft = {700,400}, \textquoteright = {700,400},
4761 <ptm>        \textquoteleft = {800,500}, \textquoteright = {800,500},
4762 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
4763 <blg>       \textquotedblright = {300,300}
4764 <cmr>       \textquotedblleft = {700,100}, \textquotedblright = {500,300}
4765 <pad>       \textquotedblleft = {700,200}, \textquotedblright = {700,200}
4766 <ppl>       \textquotedblleft = {500,300}, \textquotedblright = {500,300}
4767 <ptm>       \textquotedblleft = {700,400}, \textquotedblright = {700,400}
4768 <ugm>       \textquotedblleft = {600,200}, \textquotedblright = {600,200}
4769 }
4770
4771 <*cmr>
4772 \SetProtrusion
4773 [ name      = cmr-it-OT1,
4774   load      = cmr-it  ]
4775 { encoding = {OT1,OT4},
4776   family   = cmr,
4777   shape     = it      }
4778 {
4779   \AE = {100,  },
4780   \OE = {100,  },
4781   "00 = {200,150}, % \Gamma
4782   "01 = {150,100}, % \Delta
4783   "02 = {150, 50}, % \Theta
4784   "03 = {150, 50}, % \Lambda
4785   "04 = {100,100}, % \Xi
4786   "05 = {100,100}, % \Pi
4787   "06 = {100, 50}, % \Sigma
4788   "07 = {200,150}, % \Upsilon
4789   "08 = {150, 50}, % \Phi
4790   "09 = {150,100}, % \Psi
4791   "0A = { 50, 50} % \Omega
4792 }
4793
4794 </cmr>
4795 \SetProtrusion
4796 <m-t> [ name      = T1-it-default,
4797 <bch> [ name      = bch-it-T1,
4798 <blg> [ name      = blg-it-T1,
4799 <cmr> [ name      = cmr-it-T1,
4800 <pad> [ name      = pad-it-T1,
4801 <pmn> [ name      = pmn-it-T1,
4802 <ppl> [ name      = ppl-it-T1,
4803 <ptm> [ name      = ptm-it-T1,
4804 <ugm> [ name      = ugm-it-T1,
4805 <m-t> load      = OT1-it  ]
4806 <bch> load      = bch-it  ]
4807 <blg> load      = blg-T1  ]
4808 <cmr> load      = cmr-it  ]
4809 <pmn> load      = pmn-it  ]
4810 <pad> load      = pad-it  ]
4811 <ppl> load      = ppl-it  ]
4812 <ptm> load      = ptm-it  ]
4813 <ugm> load      = ugm-it  ]
4814 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},

```

```

4815 <blg|ptm|ugm> { encoding = {T1},
4816 <bch> family = bch,
4817 <blg> family = blg,
4818 <cmr> family = cmr,
4819 <pmn> family = pmnj,
4820 <pad> family = {pad,padx,padj},
4821 <ppl> family = {ppl,pplx,pplj},
4822 <ptm> family = {ptm,ptmx,ptmj},
4823 <ugm> family = ugm,
4824 <m-t|bch|pad|pmn|ppl|ptm> shape = {it,sl} }
4825 <blg|cmr|ugm> shape = it }
4826 {
4827 <m-t|bch|pmn> _ = { ,100},
4828 <blg> _ = {0,300},
4829 <cmr|ugm> _ = {100,200},
4830 <pad|ppl|ptm> _ = {100,100},
4831 <blg> . = {400,600},
4832 <blg> {,}= {300,500},
4833 <cmr> \AE = {100, },
4834 <bch> \OE = { 50, },
4835 <cmr> \OE = {100, },
4836 <pmn> 031 = { , -100}, % ff1
4837 <cmr|ptm> 156 = {100, }, % IJ
4838 <pad> 156 = {50, }, % IJ
4839 <pmn> 156 = {20, }, % IJ
4840 <pmn> 188 = { , -30}, % ij
4841 <pmn> \v t = { ,100},
4842 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
4843 <cmr|ugm> \textbackslash = {300,300},
4844 <bch> \textbackslash = {150,150},
4845 <pmn> \textbackslash = {100,150},
4846 <ugm> \textbar = {200,200},
4847 <cmr> \textquotedblleft = {500,300},
4848 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
4849 <blg> \textquotedbl = {300,300}, \textquotedblleft = {300,300},
4850 <blg> \textquotedblright = {300,300}, \quotedblbase = {200,600},
4851 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
4852 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4853 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4854 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
4855 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
4856 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4857 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4858 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4859 <pad> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
4860 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4861 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
4862 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
4863 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4864 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
4865 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
4866 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4867 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
4868 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4869 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
4870 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
4871 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
4872 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
4873 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
4874 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
4875 <pmn> \textvisiblespace = {100,100}
4876 }
4877

```

```

4878 <*-t|ptm>
4879 \SetProtrusion
4880 <m-t> [ name = QX-it-default,
4881 <ptm> [ name = ptm-it-QX,
4882 <m-t> load = OT1-it ]
4883 <ptm> load = ptm-it ]
4884 { encoding = {QX},
4885 <ptm> family = {ptm,ptmx,ptmj},
4886 shape = {it,sl} }
4887 {
4888 <ptm> 009 = { , 50}, % fk
4889 {=} = {100,100},
4890 <m-t> \textunderscore = {100,100},
4891 <ptm> \textunderscore = {100,150},
4892 \textbackslash = {100,200},
4893 \quotedblbase = {300,400},
4894 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
4895 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
4896 \textexclamdown = {200, }, \textquestiondown = {200, },
4897 \textbraceleft = {200,100}, \textbraceright = {200,200},
4898 \textless = {100,100}, \textgreater = {100,100},
4899 \textminus = {200,200}, \textdegree = {300,150},
4900 <m-t> \copyright = {100,100}, \textregistered = {100,100}
4901 <ptm> \textregistered = {100,150}, \copyright = {100,150},
4902 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
4903 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
4904 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
4905 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
4906 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
4907 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
4908 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
4909 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
4910 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
4911 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
4912 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
4913 <ptm> \textperthousand = { ,50}
4914 }
4915
4916 </m-t|ptm>
4917 <*-cmr|bch>
4918 \SetProtrusion
4919 <cmr> [ name = cmr-it-T5,
4920 <cmr> load = cmr-it ]
4921 <bch> [ name = bch-it-T5,
4922 <bch> load = bch-it ]
4923 { encoding = T5,
4924 <bch> family = bch,
4925 <cmr> family = cmr,
4926 shape = it }
4927 {
4928 <bch> _ = { ,100},
4929 <cmr> _ = {100,200},
4930 <bch> \textbackslash = {150,150},
4931 <cmr> \textbackslash = {300,300},
4932 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4933 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4934 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4935 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4936 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
4937 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4938 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
4939 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
4940 <bch> \textless = {100, }, \textgreater = { ,100}

```

```

4941 <cmr> \textless = {300,100}, \textgreater = {200,100}
4942 }
4943
4944 </cmr|bch>

```

Slanted is very similar to italic.

```

4945 <*cmr>
4946 \SetProtrusion
4947 [ name = cmr-sl,
4948   load = cmr-it-OT1 ]
4949 { encoding = {OT1,OT4},
4950   family = cmr,
4951   shape = sl }
4952 {
4953   L = { ,50},
4954   f = { ,-50},
4955   - = {300, },
4956   \textendash = {400, }, \textemdash = {300, }
4957 }
4958
4959 \SetProtrusion
4960 [ name = cmr-sl-T1,
4961   load = cmr-it-T1 ]
4962 { encoding = {T1,LY1},
4963   family = cmr,
4964   shape = sl }
4965 {
4966   L = { ,50},
4967   f = { ,-50},
4968   - = {300, },
4969   \textendash = {400, }, \textemdash = {300, }
4970 }
4971
4972 \SetProtrusion
4973 [ name = cmr-sl-T5,
4974   load = cmr-it-T5 ]
4975 { encoding = T5,
4976   family = cmr,
4977   shape = sl }
4978 {
4979   L = { ,50},
4980   f = { ,-50},
4981   - = {300, },
4982   \textendash = {400, }, \textemdash = {300, }
4983 }
4984
4985 \SetProtrusion
4986 [ name = lmr-it-T1,
4987   load = cmr-it-T1 ]
4988 { encoding = {T1,LY1},
4989   family = lmr,
4990   shape = {it,sl} }
4991 {
4992   \textquotedblleft = {700,100},
4993   \quotedblbase = {600,300}
4994 }
4995

```

Oldstyle numerals are slightly different.

```

4996 \SetProtrusion
4997 [ name = cmr(oldstyle)-it,
4998   load = cmr-it-T1 ]
4999 { encoding = T1,

```

```

5000     family   = {hfor,cmor},
5001     shape    = {it,sl} }
5002 {
5003     1 = {250, 50},
5004     2 = {150,-100},
5005     3 = {100,-50},
5006     4 = {150,150},
5007     6 = {200,  },
5008     7 = {200, 50},
5009     8 = {150,-50},
5010     9 = {100, 50}
5011 }
5012
5013 </cmr>
5014 < *pmn>
5015 \SetProtrusion
5016 [ name    = pmnx-it,
5017   load    = pmnj-it ]
5018 { encoding = OT1,
5019   family   = pmnx,
5020   shape    = {it,sl} }
5021 {
5022     1 = {100,150}
5023 }
5024
5025 \SetProtrusion
5026 [ name    = pmnx-it-T1,
5027   load    = pmnj-it-T1 ]
5028 { encoding = {T1,LY1},
5029   family   = pmnx,
5030   shape    = {it,sl} }
5031 {
5032     1 = {100,150}
5033 }
5034
5035 </pmn>
5036 < *ptm>
5037 \SetProtrusion
5038 [ name    = ptm-it-LY1,
5039   load    = ptm-it-T1 ]
5040 { encoding = {LY1},
5041   family   = {ptm,ptmx,ptmj},
5042   shape    = {it,sl} }
5043 {
5044     -                      = {100,100},
5045     \texttrademark         = {100,100},
5046     \textregistered        = {100,100},
5047     \textcopyright         = {100,100},
5048     \textdegree            = {300,100},
5049     \textminus             = {200,200},
5050     \textellipsis          = {100,200},
5051     \texteuro              = {  ,  },
5052     \textcent              = {100,100},
5053     \textquotesingle        = {500,  },
5054     \textflorin             = {100, 70},
5055     \textdagger            = {150,150},
5056     \textdaggerdbl         = {100,100},
5057     \textbullet            = {150,150},
5058     \textonesuperior        = {150,100},
5059     \texttwosuperior        = {150, 50},
5060     \textthreesuperior      = {150, 50},
5061     \textparagraph          = {100,  },
5062     \textperiodcentered     = {500,300},

```



```

5063 \textonequarter      = { 50,  },
5064 \textonehalf         = { 50,  },
5065 \textplusminus        = {100,100},
5066 \textmultiply        = {150,150},
5067 \textdivide          = {150,150}
5068 }
5069
5070 </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).

```

5071 <*(big|ugm)>
5072 \SetProtrusion
5073 <m-t> [ name      = OT1-sc,
5074 <bch> [ name      = bch-sc,
5075 <cmr> [ name      = cmr-sc-OT1,
5076 <pad> [ name      = pad-sc,
5077 <pmn> [ name      = pmnj-sc,
5078 <ppl> [ name      = ppl-sc,
5079 <ptm> [ name      = ptm-sc,
5080 <m-t> load      = default ]
5081 <bch> load      = bch-default ]
5082 <cmr> load      = cmr-OT1 ]
5083 <pad> load      = pad-default ]
5084 <pmn> load      = pmnj-default ]
5085 <ppl> load      = ppl-default ]
5086 <ptm> load      = ptm-default ]
5087 <m-t|bch|pad|pmn> { encoding = OT1,
5088 <cmr|ppl|ptm> { encoding = {OT1,OT4},
5089 <bch> family     = bch,
5090 <cmr> family     = cmr,
5091 <pad> family     = {pad,padx,padj},
5092 <pmn> family     = pmnj,
5093 <ppl> family     = {ppl,pplx,pplj},
5094 <ptm> family     = {ptm,ptmx,ptmj},
5095 shape          = sc }
5096 {
5097   a = {50,50},
5098 <cmr|pad|ppl|ptm> \ae = {50,  },
5099 <bch|pmn> c = {50,  },
5100 <bch|pad|pmn> d = { ,50},
5101 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
5102 <bch|pad|pmn> g = {50,  },
5103 <m-t|cmr|pad|pmn|ppl|ptm> j = {50,  },
5104 <bch> j = {100,  },
5105 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
5106 <ptm> l = { ,80},
5107 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
5108 <ptm> 013 = { ,80}, % fl
5109 <bch|pad|pmn> o = {50,50},
5110 <pad|pmn> \oe = {50,  },
5111 <ppl> p = { 0, 0},
5112 <bch|pad|pmn> q = {50,70},
5113 <ppl> q = { 0,  },
5114 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5115 t = {50,50},
5116 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5117 <ptm> y = {80,80}

```

```

5118     }
5119
5120 \SetProtrusion
5121 <m-t> [ name      = T1-sc,
5122 <bch>  [ name      = bch-sc-T1,
5123 <cmr>  [ name      = cmr-sc-T1,
5124 <pad>  [ name      = pad-sc-T1,
5125 <pmn>  [ name      = pmnj-sc-T1,
5126 <ppl>  [ name      = ppl-sc-T1,
5127 <ptm>  [ name      = ptm-sc-T1,
5128 <m-t>  load      = T1-default ]
5129 <bch>  load      = bch-T1    ]
5130 <cmr>  load      = cmr-T1    ]
5131 <pad>  load      = pad-T1    ]
5132 <pmn>  load      = pmnj-T1   ]
5133 <ppl>  load      = ppl-T1    ]
5134 <ptm>  load      = ptm-T1    ]
5135 { encoding = {T1,LY1},
5136 <bch>  family  = bch,
5137 <cmr>  family  = cmr,
5138 <pad>  family  = {pad,padx,padj},
5139 <pmn>  family  = pmnj,
5140 <ppl>  family  = {ppl,pplx,pplj},
5141 <ptm>  family  = {ptm,ptmx,ptmj},
5142 shape  = sc }
5143 {
5144   a = {50,50},
5145 <cmr|pad|ppl|ptm> \ae = {50, },
5146 <bch|pmn>         c = {50, },
5147 <bch|pad|pmn>     d = { ,50},
5148 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
5149 <bch|pad|pmn>     g = {50, },
5150 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
5151 <bch>             j = {100, },
5152 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
5153 <ptm>            l = { ,80},
5154 <m-t|bch|cmr|pad|pmn|ppl> 029 = { ,50}, % fl
5155 <ptm>            029 = { ,80}, % fl
5156 <bch|pad|pmn>     o = {50,50},
5157 <bch|pad|pmn>     \oe = {50, },
5158 <ppl>            p = { 0, 0},
5159 <bch|pad|pmn>     q = {50,70},
5160 <ppl>            q = { 0, },
5161 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5162 t = {50,50},
5163 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5164 <ptm>           y = {80,80}
5165 }
5166
5167 </(blg|ugm)>
5168 <*m-t>
5169 \SetProtrusion
5170 [ name      = QX-sc,
5171 load      = QX-default ]
5172 { encoding = QX,
5173 shape      = sc }
5174 {
5175   a = {50,50},
5176   f = { ,50},
5177   j = {50, },
5178   l = { ,50},
5179   013 = { ,50}, % fl
5180   r = { , 0},

```

```

5181     t = {50,50},
5182     y = {50,50}
5183   }
5184
5185 </m-t>
5186 <*cmr>|bch>
5187 \SetProtrusion
5188 <bch> [ name      = bch-sc-T5,
5189 <bch>   load      = bch-T5 ]
5190 <cmr> [ name      = cmr-sc-T5,
5191 <cmr>   load      = cmr-T5 ]
5192 { encoding = T5,
5193 <bch>   family   = bch,
5194 <cmr>   family   = cmr,
5195   shape    = sc }
5196 {
5197   a = {50,50},
5198 <bch>   c = {50, },
5199 <bch>   d = { ,50},
5200   f = { ,50},
5201 <bch>   g = {50, },
5202 <bch>   j = {100, },
5203 <cmr>   j = {50, },
5204   l = { ,50},
5205 <bch>   o = {50,50},
5206 <bch>   q = { 0, },
5207 <cmr>   r = { , 0},
5208   t = {50,50},
5209   y = {50,50}
5210 }
5211
5212 </cmr>|bch>
5213 <*pmn>
5214 \SetProtrusion
5215 [ name      = pmnx-sc,
5216   load      = pmnj-sc ]
5217 { encoding = OT1,
5218   family   = pmnx,
5219   shape    = sc }
5220 {
5221   l = {230,180}
5222 }
5223
5224 \SetProtrusion
5225 [ name      = pmnx-sc-T1,
5226   load      = pmnj-sc-T1 ]
5227 { encoding = {T1,LY1},
5228   family   = pmnx,
5229   shape    = sc }
5230 {
5231   l = {230,180}
5232 }
5233

```

#### 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`.

```

5234 \SetProtrusion
5235 [ name      = pmnj-scit,
5236   load      = pmnj-it ]

```

```

5237 { encoding = OT1,
5238     family   = pmnj,
5239     shape    = {scit,si} }
5240 {
5241     a = {50, },
5242     \ae = { , -50},
5243     b = {20, -50},
5244     c = {50, -50},
5245     d = {20, 0},
5246     e = {20, -50},
5247     f = {10, 0},
5248     012 = {10, -50}, % fi
5249     013 = {10, -50}, % fl
5250     014 = {10, -50}, % ffi
5251     015 = {10, -50}, % ffl
5252     g = {50, -50},
5253     i = {20, -50},
5254     j = {20, 0},
5255     k = {20, },
5256     l = {20, 50},
5257     m = { , -30},
5258     n = { , -30},
5259     o = {50, },
5260     \oe = {50, -50},
5261     p = {20, -50},
5262     q = {50, },
5263     r = {20, 0},
5264     s = {20, -30},
5265     t = {70, },
5266     u = {50, -50},
5267     v = {100, },
5268     w = {100, },
5269     y = {50, },
5270     z = { , -50}
5271 }
5272
5273 \SetProtrusion
5274 [ name   = pmnj-scit-T1,
5275   load   = pmnj-it-T1 ]
5276 { encoding = {T1,LY1},
5277   family   = pmnj,
5278   shape    = {scit,si} }
5279 {
5280     a = {50, },
5281     \ae = { , -50},
5282     b = {20, -50},
5283     c = {50, -50},
5284     d = {20, 0},
5285     e = {20, -50},
5286     f = {10, 0},
5287     028 = {10, -50}, % fi
5288     029 = {10, -50}, % fl
5289     030 = {10, -50}, % ffi
5290     031 = {10, -50}, % ffl
5291     g = {50, -50},
5292     i = {20, -50},
5293     188 = {20, 0}, % ij
5294     j = {20, 0},
5295     k = {20, },
5296     l = {20, 50},
5297     m = { , -30},
5298     n = { , -30},
5299     o = {50, },

```

```

5300 \oe = {50,-50},
5301 p = {20,-50},
5302 q = {50, },
5303 r = {20, 0},
5304 s = {20,-30},
5305 t = {70, },
5306 u = {50,-50},
5307 v = {100, },
5308 w = {100, },
5309 y = {50, },
5310 z = { , -50}
5311 }
5312
5313 \SetProtrusion
5314 [ name = pmnx-scit,
5315 load = pmnj-scit ]
5316 { encoding = OT1,
5317 family = pmnx,
5318 shape = {scit,si} }
5319 {
5320 1 = {100,150}
5321 }
5322
5323 \SetProtrusion
5324 [ name = pmnx-scit-T1,
5325 load = pmnj-scit-T1 ]
5326 { encoding = {T1,LY1},
5327 family = pmnx,
5328 shape = {scit,si} }
5329 {
5330 1 = {100,150}
5331 }
5332
5333 </pmn>

```

### 15.8.5 textcomp

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

5334 \SetProtrusion
5335 <m-t> [ name = textcomp ]
5336 <bch> [ name = bch-textcomp ]
5337 <blg> [ name = blg-textcomp ]
5338 <cmr> [ name = cmr-textcomp ]
5339 <pad> [ name = pad-textcomp ]
5340 <pmn> [ name = pmn-textcomp ]
5341 <ppl> [ name = ppl-textcomp ]
5342 <ptm> [ name = ptm-textcomp ]
5343 <ugm> [ name = ugm-textcomp ]
5344 <m-t> { encoding = TS1 }
5345 <!m-t> { encoding = TS1,
5346 <bch> family = bch }
5347 <blg> family = blg }
5348 <cmr> family = cmr }
5349 <pad> family = {pad,padx,padj} }
5350 <pmn> family = {pmnx,pmnj} }
5351 <ppl> family = {ppl,pplx,pplj} }
5352 <ptm> family = {ptm,ptmx,ptmj} }
5353 <ugm> family = ugm }
5354 {
5355 <blg> \textquotestraightbase = {400,500},

```

```

5356 <cmr> \textquotestraightbase = {300,300},
5357 <pad|pmn> \textquotestraightbase = {400,400},
5358 <blg> \textquotestraightdblbase = {300,400},
5359 <cmr|pmn> \textquotestraightdblbase = {300,300},
5360 <pad> \textquotestraightdblbase = {400,400},
5361 <bch|cmr|pad|pmn|ugm> \texttwelveudash = {200,200},
5362 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
5363 <ugm> \textthreequartersemdash = {200,200},
5364 <blg> \textquotesingle = {500,600},
5365 <cmr|pmn> \textquotesingle = {300,400},
5366 <pad> \textquotesingle = {400,500},
5367 <ptm> \textquotesingle = {500,500},
5368 <ugm> \textquotesingle = {300,500},
5369 <bch|cmr|pmn> \textasteriskcentered = {200,300},
5370 <blg> \textasteriskcentered = {150,200},
5371 <pad> \textasteriskcentered = {300,300},
5372 <ugm> \textasteriskcentered = {100,200},
5373 <pmn> \textfractionsolidus = {-200,-200},
5374 <cmr> \textoneoldstyle = {100,100},
5375 <pmn> \textoneoldstyle = { , 50},
5376 <cmr> \textthreeoldstyle = { , 50},
5377 <pad|pmn> \textthreeoldstyle = { 50, },
5378 <cmr> \textfouroldstyle = { 50, 50},
5379 <pad|pmn> \textfouroldstyle = { 50, },
5380 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
5381 <cmr> \textlangle = {400, },
5382 <cmr> \textrangle = { ,400},
5383 <m-t|bch|pmn|ptm> \textminus = {200,200},
5384 <cmr|pad|ppl> \textminus = {300,300},
5385 <blg|ugm> \textminus = {250,300},
5386 <bch|pad|pmn> \textlbrackdbl = {100, },
5387 <blg> \textlbrackdbl = {200, },
5388 <bch|pad|pmn> \textrbrackdbl = { ,100},
5389 <blg> \textrbrackdbl = { ,200},
5390 <pmn> \textasciigrave = {200,500},
5391 <bch|blg|cmr|pad|pmn> \texttildelow = {200,250},
5392 <pmn> \textasciibreve = {300,400},
5393 <pmn> \textasciicaron = {300,400},
5394 <pmn> \textacutedbl = {200,300},
5395 <pmn> \textgravedbl = {150,300},
5396 <bch|pmn|ugm> \textdagger = { 80, 80},
5397 <blg> \textdagger = {200,200},
5398 <cmr|pad> \textdagger = {100,100},
5399 <ptm> \textdagger = {150,150},
5400 <blg> \textdaggerdbl = {150,150},
5401 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
5402 <ptm> \textdaggerdbl = {100,100},
5403 <bch> \textbardbl = {100,100},
5404 <blg|ugm> \textbardbl = {150,150},
5405 <bch> \textbullet = {200,200},
5406 <blg> \textbullet = {400,500},
5407 <cmr|pad|pmn> \textbullet = { ,100},
5408 <ptm> \textbullet = {150,150},
5409 <ugm> \textbullet = { 50,100},
5410 <bch|cmr|pmn> \textcelsius = { 50, },
5411 <pad> \textcelsius = { 80, },
5412 <bch> \textflorin = { 50, 50},
5413 <blg> \textflorin = {100,100},
5414 <pad|ugm> \textflorin = { ,100},
5415 <pmn> \textflorin = { 50,100},
5416 <ptm> \textflorin = { 50, 70},
5417 <cmr> \textcolonmonetary = { , 50},
5418 <pad|pmn> \textcolonmonetary = { 50, },

```

```

5419 <pmn> \textinterrobang = { ,100},
5420 <pmn> \textinterrobangdown = {100, },
5421 <m-t|pad|ptm> \texttrademark = {100,100},
5422 <bch> \texttrademark = {150,150},
5423 <blg|cmr|ppl> \texttrademark = {200,200},
5424 <pmn> \texttrademark = { 50, 50},
5425 <ugm> \texttrademark = {100,150},
5426 <bch|ugm> \textcent = { 50, },
5427 <ptm> \textcent = {100,100},
5428 <bch> \textsterling = { 50, },
5429 <ugm> \textsterling = { , 50},
5430 <bch> \textbrokenbar = {200,200},
5431 <blg> \textbrokenbar = {250,250},
5432 <ugm> \textbrokenbar = {200,300},
5433 <pmn> \textasciidieresis = {300,400},
5434 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
5435 <pmn> \textcopyright = {100,150},
5436 <ppl> \textcopyright = {200,200},
5437 <bch|cmr|ugm> \textordfeminine = {100,200},
5438 <pad|pmn> \textordfeminine = {200,200},
5439 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
5440 <blg> \textlnot = {200,100},
5441 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
5442 <pmn> \textregistered = { 50,150},
5443 <ppl> \textregistered = {200,200},
5444 <pmn> \textasciimacron = {150,200},
5445 <m-t|ppl|ptm> \textdegree = {300,300},
5446 <bch> \textdegree = {150,200},
5447 <blg|ugm> \textdegree = {200,200},
5448 <cmr|pad> \textdegree = {400,400},
5449 <pmn> \textdegree = {150,400},
5450 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
5451 <blg> \textpm = {100,100},
5452 <ptm> \textpm = { 50, 80},
5453 <bch|blg|ugm> \texttwosuperior = {100,200},
5454 <cmr> \texttwosuperior = { 50,100},
5455 <pad|pmn> \texttwosuperior = {200,200},
5456 <ptm> \texttwosuperior = { 50, 50},
5457 <bch|blg|ugm> \textthreesuperior = {100,200},
5458 <cmr> \textthreesuperior = { 50,100},
5459 <pad|pmn> \textthreesuperior = {200,200},
5460 <ptm> \textthreesuperior = { 50, 50},
5461 <pmn> \textasciiaacute = {300,400},
5462 <bch|ugm> \textmu = { ,100},
5463 <bch|pad|pmn> \textparagraph = { ,100},
5464 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
5465 <blg> \textperiodcentered = {400,500},
5466 <ptm> \textperiodcentered = {300,300},
5467 <ugm> \textperiodcentered = {200,500},
5468 <bch|blg|ugm> \textonesuperior = {200,300},
5469 <cmr|pad|pmn> \textonesuperior = {200,200},
5470 <ptm> \textonesuperior = {100,100},
5471 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
5472 <blg|cmr> \textordmasculine = {100,200},
5473 <bch|cmr|pmn> \texteuro = {100, },
5474 <pad> \texteuro = { 50,100},
5475 <bch> \texttimes = {200,200},
5476 <blg|ptm> \texttimes = {100,100},
5477 <cmr> \texttimes = {150,250},
5478 <pad> \texttimes = {100,150},
5479 <pmn> \texttimes = { 70,100},
5480 <ugm> \texttimes = {200,300},
5481 <bch|pad|pmn> \textdiv = {150,200}

```

```

5482 <big> \textdiv = {100,100}
5483 <cmr> \textdiv = {150,250}
5484 <ptm> \textdiv = { 50,100},
5485 <ugm> \textdiv = {200,300},
5486 <ptm> \textperthousand = { ,50}
5487 <ugm> \textsection = { ,100},
5488 <ugm> \textonehalf = { 50,100},
5489 <ugm> \textonequarter = { 50,100},
5490 <ugm> \textthreequarters = { 50,100},
5491 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

5492 }
5493
5494 <*cmr|pad|pmn|ugm>
5495 \SetProtrusion
5496 <cmr> [ name = cmr-textcomp-it ]
5497 <pad> [ name = pad-textcomp-it ]
5498 <pmn> [ name = pmn-textcomp-it ]
5499 <ugm> [ name = ugm-textcomp-it ]
5500 { encoding = TS1,
5501 <cmr> family = cmr,
5502 <pad> family = {pad,padx,padj},
5503 <pmn> family = {pmnx,pmnj},
5504 <ugm> family = ugm,
5505 <!ugm> shape = {it,sl} }
5506 <ugm> shape = it }
5507 {
5508 <cmr> \textquotestraightbase = {300,600},
5509 <pad|pmn> \textquotestraightbase = {400,400},
5510 <cmr> \textquotestraightdblbase = {300,600},
5511 <pad> \textquotestraightdblbase = {300,400},
5512 <pmn> \textquotestraightdblbase = {300,300},
5513 \texttwelveudash = {200,200},
5514 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
5515 <ugm> \textthreequartersemdash = {200,200},
5516 <cmr> \textquotesingle = {600,300},
5517 <pad> \textquotesingle = {800,100},
5518 <pmn> \textquotesingle = {300,200},
5519 <ugm> \textquotesingle = {500,500},
5520 <cmr> \textasteriskcentered = {300,200},
5521 <pad> \textasteriskcentered = {500,100},
5522 <pmn> \textasteriskcentered = {200,300},
5523 <ugm> \textasteriskcentered = {300,150},
5524 <pmn> \textfractionsolidus = {-200,-200},
5525 <cmr> \textoneoldstyle = {100, 50},
5526 <pad> \textoneoldstyle = {100, },
5527 <pmn> \textoneoldstyle = { 50, },
5528 <pad> \texttwooldstyle = { 50, },
5529 <pmn> \texttwooldstyle = {-50, },
5530 <cmr> \textthreeoldstyle = {100, 50},
5531 <pmn> \textthreeoldstyle = {-100, },
5532 <cmr> \textfouroldstyle = { 50, 50},
5533 <pad> \textfouroldstyle = { 50,100},
5534 <cmr> \textsevenoldstyle = { 50, 80},
5535 <pad> \textsevenoldstyle = { 50, },
5536 <pmn> \textsevenoldstyle = { 20, },
5537 <cmr> \textlangle = {400, },
5538 <cmr> \textrangle = { ,400},
5539 <cmr|pad> \textminus = {300,300},
5540 <pmn> \textminus = {200,200},
5541 <ugm> \textminus = {250,300},
5542 <pad|pmn> \textlbrackdbl = {100, },

```



```

5543 <pad|pmn> \textrbrackdbl = { ,100},
5544 <pmn> \textasciigrave = {300,300},
5545 <cmr|pad|pmn> \texttildelow = {200,250},
5546 <pmn> \textasciibreve = {300,300},
5547 <pmn> \textasciicaron = {300,300},
5548 <pmn> \textacutedbl = {200,300},
5549 <pmn> \textgravedbl = {150,300},
5550 <cmr> \textdagger = {100,100},
5551 <pad> \textdagger = {200,100},
5552 <pmn> \textdagger = { 80, 50},
5553 <ugm> \textdagger = { 80, 80},
5554 <cmr|pad> \textdaggerdbl = { 80, 80},
5555 <pmn> \textdaggerdbl = { 80, 50},
5556 <ugm> \textbardbl = {150,150},
5557 <cmr> \textbullet = {200,100},
5558 <pad> \textbullet = {300, },
5559 <pmn> \textbullet = { 30, 70},
5560 <ugm> \textbullet = { 50,100},
5561 <cmr> \textcelsius = {100, },
5562 <pad> \textcelsius = {200, },
5563 <pmn> \textcelsius = { 50,-50},
5564 <pad> \textflorin = {100, },
5565 <pmn> \textflorin = { 50,100},
5566 <ugm> \textflorin = { ,100},
5567 <cmr> \textcolonmonetary = {150, },
5568 <pad> \textcolonmonetary = {100, },
5569 <pmn> \textcolonmonetary = { 50,-50},
5570 <cmr|pad> \texttrademark = {200, },
5571 <pmn> \texttrademark = { 50,100},
5572 <ugm> \texttrademark = {150, 50},
5573 <ugm> \textcent = { 50, },
5574 <ugm> \textsterling = { , 50},
5575 <ugm> \textbrokenbar = {200,300},
5576 <pmn> \textasciidieresis = {300,200},
5577 <cmr> \textcopyright = {100, },
5578 <pad> \textcopyright = {200,100},
5579 <pmn> \textcopyright = {100,150},
5580 <ugm> \textcopyright = {300, },
5581 <cmr> \textordfeminine = {100,100},
5582 <pmn> \textordfeminine = {200,200},
5583 <ugm> \textordfeminine = {100,200},
5584 <cmr|pad> \textlnot = {300, },
5585 <pmn|ugm> \textlnot = {200, },
5586 <cmr> \textregistered = {100, },
5587 <pad> \textregistered = {200,100},
5588 <pmn> \textregistered = { 50,150},
5589 <ugm> \textregistered = {300, },
5590 <pmn> \textasciimacron = {150,200},
5591 <cmr|pad> \textdegree = {500,100},
5592 <pmn> \textdegree = {150,150},
5593 <ugm> \textdegree = {300,200},
5594 <cmr> \textpm = {150,100},
5595 <pad> \textpm = {200,150},
5596 <pmn|ugm> \textpm = {150,200},
5597 <cmr> \textonesuperior = {400, },
5598 <pad> \textonesuperior = {300,100},
5599 <pmn> \textonesuperior = {200,100},
5600 <ugm> \textonesuperior = {300,300},
5601 <cmr> \texttwosuperior = {400, },
5602 <pad> \texttwosuperior = {300, },
5603 <pmn> \texttwosuperior = {200,100},
5604 <ugm> \texttwosuperior = {300,200},
5605 <cmr> \textthreesuperior = {400, },

```

```

5606 <pad> \textthreesuperior = {300, },
5607 <pmn> \textthreesuperior = {200,100},
5608 <ugm> \textthreesuperior = {300,200},
5609 <ugm> \textmu = { ,100},
5610 <pmn> \textasciicute = {300,200},
5611 <cmr> \textparagraph = {200, },
5612 <pmn> \textparagraph = { ,100},
5613 <cmr> \textperiodcentered = {500,500},
5614 <pad|pmn|ugm> \textperiodcentered = {300,400},
5615 <cmr> \textordmasculine = {100,100},
5616 <pmn> \textordmasculine = {200,200},
5617 <ugm> \textordmasculine = {300,200},
5618 <cmr> \texteuro = {200, },
5619 <pad> \texteuro = {100, },
5620 <pmn> \texteuro = {100,-50},
5621 <cmr> \texttimes = {200,200},
5622 <pad> \texttimes = {200,100},
5623 <pmn> \texttimes = { ,70,100},
5624 <ugm> \texttimes = {200,300},
5625 <cmr|pad> \textdiv = {200,200},
5626 <pmn> \textdiv = {150,200},
5627 <ugm> \textdiv = {200,300},
5628 <ugm> \textsection = { ,200},
5629 <ugm> \textonehalf = { 50,100},
5630 <ugm> \textonequarter = { 50,100},
5631 <ugm> \textthreequarters = { 50,100},
5632 <ugm> \textsurd = { ,100}
5633 }
5634
5635 </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}

```

```

5636 <*cmr>
5637 \SetProtrusion
5638 [ name = cmr-math-letters ]
5639 { encoding = OML,
5640   family = cmm,
5641   series = {m,b},
5642   shape = it }
5643 {
5644   A = {100, 50}, % \mathnormal
5645   B = { 50, },
5646   C = { 50, },

```

```

5647     D = { 50, 50},
5648     E = { 50,   },
5649     F = {100, 50},
5650     G = { 50, 50},
5651     H = { 50, 50},
5652     I = { 50, 50},
5653     J = {150, 50},
5654     K = { 50,100},
5655     L = { 50, 50},
5656     M = { 50,   },
5657     N = { 50,   },
5658     O = { 50,   },
5659     P = { 50,   },
5660     Q = { 50, 50},
5661     R = { 50,   },
5662     S = { 50,   },
5663     T = { 50,100},
5664     U = { 50, 50},
5665     V = {100,100},
5666     W = { 50,100},
5667     X = { 50,100},
5668     Y = {100,100},
5669     f = {100,100},
5670     h = {   ,100},
5671     i = {   , 50},
5672     j = {   , 50},
5673     k = {   , 50},
5674     r = {   , 50},
5675     v = {   , 50},
5676     w = {   , 50},
5677     x = {   , 50},
5678     "0B = { 50,100}, % \alpha
5679     "0C = { 50, 50}, % \beta
5680     "0D = {200,150}, % \gamma
5681     "0E = { 50, 50}, % \delta
5682     "0F = { 50, 50}, % \epsilon
5683     "10 = { 50,150}, % \zeta
5684     "12 = { 50,   }, % \theta
5685     "13 = {   ,100}, % \iota
5686     "14 = {   ,100}, % \kappa
5687     "15 = {100, 50}, % \lambda
5688     "16 = {   , 50}, % \mu
5689     "17 = {   , 50}, % \nu
5690     "18 = {   , 50}, % \xi
5691     "19 = { 50,100}, % \pi
5692     "1A = { 50, 50}, % \rho
5693     "1B = {   ,150}, % \sigma
5694     "1C = { 50,150}, % \tau
5695     "1D = { 50, 50}, % \upsilon
5696     "1F = { 50,100}, % \chi
5697     "20 = { 50, 50}, % \psi
5698     "21 = {   , 50}, % \omega
5699     "22 = {   , 50}, % \varepsilon
5700     "23 = {   , 50}, % \vartheta
5701     "24 = {   , 50}, % \varpi
5702     "25 = {100,   }, % \varrho
5703     "26 = {100,100}, % \varsigma
5704     "27 = { 50, 50}, % \varphi
5705     "28 = {100,100}, % \leftharpoonup
5706     "29 = {100,100}, % \leftharpoondown
5707     "2A = {100,100}, % \rightharpoonup
5708     "2B = {100,100}, % \rightharpoondown
5709     "2C = {300,200}, % \hook

```

```

5710 "2D = {200,300}, % \rhook
5711 "2E = { ,100}, % \triangleright
5712 "2F = {100, }, % \triangleleft
5713 "3A = { ,500}, % ., \ldotp
5714 "3B = { ,500}, % ,
5715 "3C = {200,100}, % <
5716 "3D = {300,400}, % /
5717 "3E = {100,200}, % >
5718 "3F = {200,200}, % \star
5719 "5B = { ,100}, % \flat
5720 "5E = {200,200}, % \smile
5721 "5F = {200,200}, % \frown
5722 "7C = {100, }, % \jmath
5723 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

5724 }
5725

```

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}

```

```

5726 \SetProtrusion
5727 [ name = cmr-math-symbols ]
5728 { encoding = OMS,
5729   family = cmsy,
5730   series = {m,b},
5731   shape = n }
5732 {
5733   A = {150, 50}, % \mathcal
5734   C = { ,100},
5735   D = { , 50},
5736   F = { 50,150},
5737   I = { ,100},
5738   J = {100,150},
5739   K = { ,100},
5740   L = {100, },
5741   M = { 50, 50},
5742   N = { 50,100},
5743   P = { , 50},
5744   Q = { 50, },
5745   R = { , 50},
5746   T = { 50,150},
5747   V = { 50, 50},
5748   W = { , 50},
5749   X = {100,100},
5750   Y = {100, },
5751   Z = {100,150},
5752   "00 = {300,300}, % -
5753   "01 = { ,700}, % \cdot, \cdotp
5754   "02 = {150,250}, % \times
5755   "03 = {150,250}, % *, \ast
5756   "04 = {200,300}, % \div
5757   "05 = {150,250}, % \diamond
5758   "06 = {200,200}, % \pm
5759   "07 = {200,200}, % \mp
5760   "08 = {100,100}, % \oplus
5761   "09 = {100,100}, % \ominus
5762   "0A = {100,100}, % \otimes
5763   "0B = {100,100}, % \oslash
5764   "0C = {100,100}, % \odot

```

```

5765 "0D = {100,100}, % \bigcirc
5766 "0E = {100,100}, % \circ
5767 "0F = {100,100}, % \bullet
5768 "10 = {100,100}, % \asymp
5769 "11 = {100,100}, % \equiv
5770 "12 = {200,100}, % \subseteq
5771 "13 = {100,200}, % \supseteq
5772 "14 = {200,100}, % \leq
5773 "15 = {100,200}, % \geq
5774 "16 = {200,100}, % \preceq
5775 "17 = {100,200}, % \succeq
5776 "18 = {200,200}, % \sim
5777 "19 = {150,150}, % \approx
5778 "1A = {200,100}, % \subset
5779 "1B = {100,200}, % \supset
5780 "1C = {200,100}, % \ll
5781 "1D = {100,200}, % \gg
5782 "1E = {300,100}, % \prec
5783 "1F = {100,300}, % \succ
5784 "20 = {100,200}, % \leftarrow
5785 "21 = {200,100}, % \rightarrow
5786 "22 = {100,100}, % \uparrow
5787 "23 = {100,100}, % \downarrow
5788 "24 = {100,100}, % \leftrightarrows
5789 "25 = {100,100}, % \nearrow
5790 "26 = {100,100}, % \searrow
5791 "27 = {100,100}, % \simeq
5792 "28 = {100,100}, % \Leftarrow
5793 "29 = {100,100}, % \Rightarrow
5794 "2A = {100,100}, % \Uparrow
5795 "2B = {100,100}, % \Downarrow
5796 "2C = {100,100}, % \Leftrightarrow
5797 "2D = {100,100}, % \nrightarrow
5798 "2E = {100,100}, % \swarrow
5799 "2F = { ,100}, % \propto
5800 "30 = { ,400}, % \prime
5801 "31 = {100,100}, % \infty
5802 "32 = {150,100}, % \in
5803 "33 = {100,150}, % \ni
5804 "34 = {100,100}, % \triangle, \bigtriangleup
5805 "35 = {100,100}, % \bigtriangledown
5806 "38 = { ,100}, % \forall
5807 "39 = {100, }, % \exists
5808 "3A = {200, }, % \neg
5809 "3E = {200,200}, % \top
5810 "3F = {200,200}, % \bot, \perp
5811 "5E = {100,200}, % \wedge
5812 "5F = {100,200}, % \vee
5813 "60 = { ,300}, % \vdash
5814 "61 = {300, }, % \dashv
5815 "62 = {100,100}, % \lfloor
5816 "63 = {100,100}, % \rfloor
5817 "64 = {100,100}, % \lceil
5818 "65 = {100,100}, % \rceil
5819 "66 = {150, }, % \lbrace
5820 "67 = { ,150}, % \rbrace
5821 "68 = {400, }, % \langle
5822 "69 = { ,400}, % \rangle
5823 "6C = {100,100}, % \updownarrow
5824 "6D = {100,100}, % \Updownarrow
5825 "6E = {100,300}, % \, \backslash, \setminus
5826 "72 = {100,100}, % \nabla
5827 "79 = {200,200}, % \dagger

```

```

5828 "7A = {100,100}, % \ddagger
5829 "7B = {100, }, % \mathparagraph
5830 "7C = {100,100}, % \clubsuit
5831 "7D = {100,100}, % \diamondsuit
5832 "7E = {100,100}, % \heartsuit
5833 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

5834 }
5835

```

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}
```

```

5836 </cmr>
5837 </cfg-t>

```

### 15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
5838 <*cfg-u>
```

Symbol font 'a'.

```

5839 <*msa>
5840 \SetProtrusion
5841 [ name = AMS-a ]
5842 { encoding = U,
5843   family = msa }
5844 {
5845   "05 = {150,250}, % \centerdot
5846   "06 = {100,100}, % \lozenge
5847   "07 = { 50, 50}, % \blacklozenge
5848   "08 = { 50, 50}, % \circlearrowright
5849   "09 = { 50, 50}, % \circlearrowleft
5850   "0A = {100,100}, % \rightleftharpoons
5851   "0B = {100,100}, % \leftrightharpoons
5852   "0D = {-50,200}, % \Vdash
5853   "0E = {-50,200}, % \Vvdash
5854   "0F = {-70,150}, % \vDash
5855   "10 = {100,150}, % \twoheadrightarrow
5856   "11 = {100,150}, % \twoheadleftarrow
5857   "12 = { 50,100}, % \leftleftarrows
5858   "13 = { 50, 80}, % \rightrightarrows
5859   "14 = {120,120}, % \upuparrows
5860   "15 = {120,120}, % \downdownarrows
5861   "16 = {200,200}, % \upharpoonright
5862   "17 = {200,200}, % \downharpoonright
5863   "18 = {200,200}, % \upharpoonleft
5864   "19 = {200,200}, % \downharpoonleft
5865   "1A = { 80,100}, % \rightarrowtail
5866   "1B = { 80,100}, % \leftarrowtail
5867   "1C = { 50, 50}, % \leftrightarrows
5868   "1D = { 50, 50}, % \rightleftarrows
5869   "1E = {250, }, % \Lsh
5870   "1F = { ,250}, % \Rsh
5871   "20 = {100,100}, % \rightsquigarrow
5872   "21 = {100,100}, % \leftrightsquigarrow
5873   "22 = {100, 50}, % \looparrowleft
5874   "23 = { 50,100}, % \looparrowright

```

```

5875 "24 = { 50, 80}, % \circeq
5876 "25 = { ,100}, % \succsim
5877 "26 = { ,100}, % \gtrsim
5878 "27 = { ,100}, % \gtrapprox
5879 "28 = {150, 50}, % \multimap
5880 "2B = {100,150}, % \doteqdot
5881 "2C = {100,150}, % \triangleq
5882 "2D = {100, 50}, % \precsim
5883 "2E = {100, 50}, % \lesssim
5884 "2F = { 50, 50}, % \lessapprox
5885 "30 = {100, 50}, % \eqslantless
5886 "31 = { 50, 50}, % \eqslantgtr
5887 "32 = {100, 50}, % \curlyeqprec
5888 "33 = { 50,100}, % \curlyeqsucc
5889 "34 = {100, 50}, % \preccurlyeq
5890 "36 = { 50, }, % \eqslant
5891 "38 = { , 50}, % \backprime
5892 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
5893 "3C = { 50,100}, % \succcurlyeq
5894 "3E = { , 50}, % \geqslant
5895 "40 = { , 50}, % \sqsubset
5896 "41 = { 50, }, % \sqsupset
5897 "42 = { ,150}, % \vartriangleright, \rhd
5898 "43 = {150, }, % \vartriangleleft, \lhd
5899 "44 = { ,100}, % \trianglerighteq, \unrhd
5900 "45 = {100, }, % \trianglelefteq, \unlhd
5901 "46 = {100,100}, % \bigstar
5902 "48 = { 50, 50}, % \blacktriangledown
5903 "49 = { ,100}, % \blacktriangleright
5904 "4A = {100, }, % \blacktriangleleft
5905 "4B = { ,150}, % \dashrightarrow (the arrow)
5906 "4C = {150, }, % \dashleftarrow
5907 "4D = { 50, 50}, % \vartriangle
5908 "4E = { 50, 50}, % \blacktriangle
5909 "4F = { 50, 50}, % \triangledown
5910 "50 = { 50, 50}, % \eqcirc
5911 "56 = { ,150}, % \Rrightarrow
5912 "57 = {150, }, % \Lleftarrow
5913 "58 = {100,300}, % \checkmark
5914 "5C = { 50, 50}, % \angle
5915 "5D = { 50, 50}, % \measuredangle
5916 "5E = { 50, 50}, % \sphericalangle
5917 "5F = { , 50}, % \varpropto
5918 "60 = {100,100}, % \smallsmile
5919 "61 = {100,100}, % \smallfrown
5920 "62 = { 50, }, % \Subset
5921 "63 = { , 50}, % \Supset
5922 "66 = {150,150}, % \curlywedge
5923 "67 = {150,150}, % \curlyvee
5924 "68 = { 50,150}, % \leftthreetimes
5925 "69 = {100, 50}, % \rightthreetimes
5926 "6C = { 50, 50}, % \bumpeq
5927 "6D = { 50, 50}, % \Bumpeq
5928 "6E = {100, }, % \lll
5929 "6F = { ,100}, % \ggg
5930 "70 = { 50,100}, % \ulcorner
5931 "71 = {100, 50}, % \urcorner
5932 "75 = {150,200}, % \dotplus
5933 "76 = { 50,100}, % \backsimeq
5934 "78 = { 50,100}, % \llcorner
5935 "79 = {100, 50}, % \lrcorner
5936 "7C = {100,100}, % \intercal
5937 "7D = { 50, 50}, % \circledcirc

```

```

5938 "7E = { 50, 50}, % \circledast
5939 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

5940 }
5941
5942 </msa>

```

Symbol font ‘b’.

```

5943 <*msb>
5944 \SetProtrusion
5945 [ name = AMS-b ]
5946 { encoding = U,
5947   family = msb }
5948 {
5949   A = { 50, 50}, % \mathbb
5950   C = { 50, 50},
5951   G = { , 50},
5952   L = { , 50},
5953   P = { , 50},
5954   R = { , 50},
5955   T = { , 50},
5956   V = { 50, 50},
5957   X = { 50, 50},
5958   Y = { 50, 50},
5959   "00 = { 50, 50}, % \lvertneqq
5960   "01 = { 50, 50}, % \gvertneqq
5961   "02 = { 50, 50}, % \nleq
5962   "03 = { 50, 50}, % \ngeq
5963   "04 = {100, 50}, % \less
5964   "05 = { 50,150}, % \ngtr
5965   "06 = {100, 50}, % \nprec
5966   "07 = { 50,150}, % \nsucc
5967   "08 = { 50, 50}, % \lneqq
5968   "09 = { 50, 50}, % \gneqq
5969   "0A = {100,100}, % \nleqslant
5970   "0B = {100,100}, % \ngeqslant
5971   "0C = {100, 50}, % \lneq
5972   "0D = { 50,100}, % \gneq
5973   "0E = {100, 50}, % \npreceq
5974   "0F = { 50,100}, % \nsucceq
5975   "10 = { 50, }, % \precnsim
5976   "11 = { 50, 50}, % \succnsim
5977   "12 = { 50, 50}, % \lnsim
5978   "13 = { 50, 50}, % \gnsim
5979   "14 = { 50, 50}, % \lneqq
5980   "15 = { 50, 50}, % \ngeqq
5981   "16 = { 50, 50}, % \precneqq
5982   "17 = { 50, 50}, % \succneqq
5983   "18 = { 50, 50}, % \precnapprox
5984   "19 = { 50, 50}, % \succnapprox
5985   "1A = { 50, 50}, % \lnapprox
5986   "1B = { 50, 50}, % \gnapprox
5987   "1C = {150,200}, % \nsim
5988   "1D = { 50, 50}, % \ncong
5989   "1E = {100,150}, % \diagup
5990   "1F = {100,150}, % \diagdown
5991   "20 = {100, 50}, % \varsubsetneq
5992   "21 = { 50,100}, % \varsupsetneq
5993   "22 = {100, 50}, % \nsubseteqq
5994   "23 = { 50,100}, % \nsupseteqq
5995   "24 = {100, 50}, % \subsetneqq
5996   "25 = { 50,100}, % \supsetneqq

```



```

5997 "26 = {100, 50}, % \varsubsetneqq
5998 "27 = { 50,100}, % \varsupsetneqq
5999 "28 = {100, 50}, % \subsetneq
6000 "29 = { 50,100}, % \supsetneq
6001 "2A = {100, 50}, % \nsubseteq
6002 "2B = { 50,100}, % \nsupseteq
6003 "2C = { 50,100}, % \nparallel
6004 "2D = {100,150}, % \nmid
6005 "2E = {150,150}, % \nshortmid
6006 "2F = {100,100}, % \nshortparallel
6007 "30 = { ,150}, % \nvdash
6008 "31 = { ,150}, % \nVdash
6009 "32 = { ,100}, % \nvDash
6010 "33 = { ,100}, % \nVDash
6011 "34 = { ,100}, % \ntrianglerighteq
6012 "35 = {100, }, % \trianglelefteq
6013 "36 = {100, }, % \triangleleft
6014 "37 = { ,100}, % \triangleright
6015 "38 = {100,200}, % \leftarrow
6016 "39 = {100,200}, % \rightarrow
6017 "3A = {100,100}, % \Leftarrow
6018 "3B = { 50,100}, % \Rightarrow
6019 "3C = {100,100}, % \Leftrightarrow
6020 "3D = {100,200}, % \leftrightarrows
6021 "3E = { 50, 50}, % \divideontimes
6022 "3F = { 50, 50}, % \varnothing
6023 "60 = {200, }, % \Finv
6024 "61 = { , 50}, % \Game
6025 "68 = {100,100}, % \eqsim
6026 "69 = { 50, }, % \beth
6027 "6A = { 50, }, % \gimel
6028 "6B = {150, }, % \daleth
6029 "6C = {200, }, % \lessdot
6030 "6D = { ,200}, % \gtrdot
6031 "6E = {100,200}, % \ltimes
6032 "6F = {150,100}, % \rtimes
6033 "70 = { 50,100}, % \shortmid
6034 "71 = { 50, 50}, % \shortparallel
6035 "72 = {200,300}, % \smallsetminus
6036 "73 = {100,200}, % \thicksim
6037 "74 = { 50,100}, % \thickapprox
6038 "75 = { 50, 50}, % \approx
6039 "76 = { 50,100}, % \succapprox
6040 "77 = { 50, 50}, % \precapprox
6041 "78 = {100,100}, % \curvearrowleft
6042 "79 = { 50,150}, % \curvearrowright
6043 "7A = { 50,200}, % \digamma
6044 "7B = {100, 50}, % \varkappa
6045 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

6046 }
6047
6048 </msb>

```

### 15.8.8 Euler

Euler Roman font (package `euler`).

```

6049 <*eur>
6050 \SetProtrusion
6051 [ name = euler ]

```

```

6052 { encoding = U,
6053     family  = eur }
6054 {
6055     "01 = {100,100},
6056     "03 = {100,150},
6057     "06 = {    ,100},
6058     "07 = {100,150},
6059     "08 = {100,100},
6060     "0A = {100,100},
6061     "0B = {    , 50},
6062     "0C = {    ,100},
6063     "0D = {100,100},
6064     "0E = {    ,100},
6065     "0F = {100,100},
6066     "10 = {100,100},
6067     "13 = {    ,100},
6068     "14 = {    ,100},
6069     "15 = {    , 50},
6070     "16 = {    , 50},
6071     "17 = { 50,100},
6072     "18 = { 50,100},
6073     "1A = {    , 50},
6074     "1B = {    , 50},
6075     "1C = { 50,100},
6076     "1D = { 50,100},
6077     "1E = { 50,100},
6078     "1F = { 50,100},
6079     "20 = {    , 50},
6080     "21 = {    , 50},
6081     "22 = { 50,100},
6082     "24 = {    , 50},
6083     "27 = { 50,100},
6084     1 = {100,100},
6085     7 = { 50,100},
6086     "3A = {300,500},
6087     "3B = {200,400},
6088     "3C = {200,100},
6089     "3D = {200,200},
6090     "3E = {100,200},
6091     A = {    ,100},
6092     D = {    , 50},
6093     J = { 50,    },
6094     K = {    , 50},
6095     L = {    , 50},
6096     Q = {    , 50},
6097     T = { 50,    },
6098     X = { 50, 50},
6099     Y = { 50,    },
6100     h = {    , 50},
6101     k = {    , 50}
6102 }
6103

```

Extended by the `eulervm` package.

```

6104 \SetProtrusion
6105 [ name      = euler-vm,
6106   load      = euler ]
6107 { encoding = U,
6108   family  = zeur }
6109 {
6110     "28 = {100,200},
6111     "29 = {100,200},
6112     "2A = {100,150},

```

```

6113     "2B = {100,150},
6114     "2C = {200,300},
6115     "2D = {200,300},
6116     "2E = {    ,100},
6117     "2F = {100,   },
6118     "3F = {150,150},
6119     "5B = {    ,100},
6120     "5E = {100,100},
6121     "5F = {100,100},
6122     "80 = {    , 50},
6123     "81 = {200,250},
6124     "82 = {100,200}
6125 }
6126
6127 </eur>

```

Euler Script font (euca1).

```

6128 <*eus>
6129 \SetProtrusion
6130 [ name      = euscript ]
6131 { encoding = U,
6132   family   = eus   }
6133 {
6134     A = {100,100},
6135     B = { 50,100},
6136     C = { 50, 50},
6137     D = { 50,100},
6138     E = { 50,100},
6139     F = { 50,   },
6140     G = { 50,   },
6141     H = {    ,100},
6142     K = {    , 50},
6143     L = {    ,150},
6144     M = {    , 50},
6145     N = {    , 50},
6146     O = { 50, 50},
6147     P = { 50, 50},
6148     T = {    ,100},
6149     U = {    , 50},
6150     V = { 50, 50},
6151     W = { 50, 50},
6152     X = { 50, 50},
6153     Y = { 50,   },
6154     Z = { 50,100},
6155     "00 = {250,250},
6156     "18 = {200,200},
6157     "3A = {200,150},
6158     "40 = {    ,100},
6159     "5E = {100,100},
6160     "5F = {100,100},
6161     "66 = { 50,   },
6162     "67 = {    , 50},
6163     "6E = {200,200}
6164 }
6165
6166 \SetProtrusion
6167 [ name      = euscript-vm,
6168   load      = euscript ]
6169 { encoding = U,
6170   family   = zeus   }
6171 {
6172     "01 = {600,600},
6173     "02 = {200,200},

```

```
6174 "03 = {200,200},
6175 "04 = {200,200},
6176 "05 = {150,150},
6177 "06 = {200,200},
6178 "07 = {200,200},
6179 "08 = {100,100},
6180 "09 = {100,100},
6181 "0A = {100,100},
6182 "0B = {100,100},
6183 "0C = {100,100},
6184 "0D = {100,100},
6185 "0E = {150,150},
6186 "0F = {100,100},
6187 "10 = {150,150},
6188 "11 = {100,100},
6189 "12 = {150,100},
6190 "13 = {100,150},
6191 "14 = {150,100},
6192 "15 = {100,150},
6193 "16 = {200,100},
6194 "17 = {100,200},
6195 "19 = {150,150},
6196 "1A = {150,100},
6197 "1B = {100,150},
6198 "1C = {100,100},
6199 "1D = {100,100},
6200 "1E = {250,100},
6201 "1F = {100,250},
6202 "20 = {150,200},
6203 "21 = {150,200},
6204 "22 = {150,150},
6205 "23 = {150,150},
6206 "24 = {100,200},
6207 "25 = {150,150},
6208 "26 = {150,150},
6209 "27 = {100,100},
6210 "28 = {100,100},
6211 "29 = {100,150},
6212 "2A = {100,100},
6213 "2B = {100,100},
6214 "2C = {100,100},
6215 "2D = {150,150},
6216 "2E = {150,150},
6217 "2F = {100,100},
6218 "30 = {100,100},
6219 "31 = {100,100},
6220 "32 = {100,100},
6221 "33 = {100,100},
6222 "34 = {100,100},
6223 "35 = {100,100},
6224 "3E = {150,150},
6225 "3F = {150,150},
6226 "60 = { ,200},
6227 "61 = {200, },
6228 "62 = {100,100},
6229 "63 = {100,100},
6230 "64 = {100,100},
6231 "65 = {100,100},
6232 "68 = {300, },
6233 "69 = { ,300},
6234 "6C = {100,100},
6235 "6D = {100,100},
6236 "6F = {100,100},
```

```

6237 "72 = {100,100},
6238 "73 = {200,100},
6239 "76 = { ,100},
6240 "77 = {100, },
6241 "78 = { 50, 50},
6242 "79 = {100,100},
6243 "7A = {100,100},
6244 "7D = {150,150},
6245 "7E = {100,100},
6246 "A8 = {100,100},
6247 "A9 = {100,100},
6248 "AB = {200,200},
6249 "BA = { ,200},
6250 "BB = { ,200},
6251 "BD = {200,200},
6252 "DE = {200,200}
6253 }
6254
6255 </eus>

```

Euler Fraktur font (eufrak).

```

6256 <*euf>
6257 \SetProtrusion
6258 [ name = mathfrak ]
6259 { encoding = U,
6260   family = euf }
6261 {
6262   A = { , 50},
6263   B = { , 50},
6264   C = { 50, 50},
6265   D = { , 80},
6266   E = { 50, },
6267   G = { , 50},
6268   L = { , 80},
6269   O = { , 50},
6270   T = { , 80},
6271   X = { 80, 50},
6272   Z = { 80, 50},
6273   b = { , 50},
6274   c = { , 50},
6275   k = { , 50},
6276   p = { , 50},
6277   q = { 50, },
6278   v = { , 50},
6279   w = { , 50},
6280   x = { , 50},
6281   1 = {100,100},
6282   2 = { 80, 80},
6283   3 = { 80, 50},
6284   4 = { 80, 50},
6285   7 = { 50, 50},
6286 "12 = {500,500},
6287 "13 = {500,500},
6288 ! = { ,200},
6289 ' = {200,300},
6290 ( = {200, },
6291 ) = { ,200},
6292 * = {200,200},
6293 + = {200,250},
6294 - = {200,200},
6295 {,} = {300,300},
6296 . = {400,400},
6297 {=} = {200,200},

```

```

6298      : = { ,200},
6299      ; = { ,200},
6300      ] = { ,200}
6301    }
6302
6303 </euf>
6304 </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>).

```

6305 <*cfg-e>
6306 \SetProtrusion
6307 <zpeu|euroitc> { encoding = U,
6308 <mvs> { encoding = {OT1,U},
6309 <zpeu> family = zpeu }
6310 <euroitc> family = {euroitc,euroitcs} }
6311 <mvs> family = mvs }
6312 {
6313 <zpeu> E = {50, }
6314 <euroitc> E = {100,50}
6315 <mvs> 164 = {50,50}, % \EUR
6316 <mvs> 068 = {50,-100} % \EURdig
6317 }
6318
6319 <*zpeu|euroitc>
6320 \SetProtrusion
6321 { encoding = U,
6322 <zpeu> family = zpeu,
6323 <euroitc> family = {euroitc,euroitcs},
6324 shape = it* }
6325 {
6326 <zpeu> E = {100,-50}
6327 <euroitc> E = {100,}
6328 }
6329
6330 </zpeu|euroitc>
6331 <*zpeu>
6332 \SetProtrusion
6333 { encoding = U,
6334 family = {zpeus,eurosans} }
6335 {
6336 E = {100,50}
6337 }
6338
6339 \SetProtrusion
6340 { encoding = U,
6341 family = {zpeus,eurosans},
6342 shape = it* }
6343 {
6344 E = {200, }
6345 }
6346
6347 </zpeu>
6348 </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.

```

6349 (*m-t)
6350 %%% -----
6351 %%% INTERWORD SPACING
6352
6353 \SetExtraSpacing
6354 [ name = default ]
6355 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6356 {

```

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

```

6357 {,} = { , -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)

```

6358 r = { , -300,300},

```

- [before or] after lowercase characters with ascenders

```

6359 b = { , -200,200},
6360 d = { , -200,200},
6361 f = { , -200,200},
6362 h = { , -200,200},
6363 k = { , -200,200},
6364 l = { , -200,200},
6365 t = { , -200,200},

```

- [before or] after lowercase characters with x-height plus descender with additional optical space, e. g., ‘v’, or ‘w’

```

6366 c = { , -100,100},

```

```

6367      p = { , -100, 100 },
6368      v = { , -100, 100 },
6369      w = { , -100, 100 },
6370      z = { , -100, 100 },
6371      x = { , -100, 100 },
6372      y = { , -100, 100 },

```

- [before or] after lowercase characters with x-height plus descender without additional optical space

```

6373      i = { , 50, -50 },
6374      m = { , 50, -50 },
6375      n = { , 50, -50 },
6376      u = { , 50, -50 },

```

- after colon and semicolon

```

6377      : = { , 200, -200 },
6378      ; = { , 200, -200 },

```

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

```

6379      . = { , 250, -250 },
6380      ! = { , 250, -250 },
6381      ? = { , 250, -250 }

```

The order has to be reversed when enlarging is needed.'

```

6382    }
6383

```

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.

```

6384 \SetExtraSpacing
6385 [ name = nonfrench-cmr,
6386   load = default,
6387   context = nonfrench ]
6388 { encoding = {OT1,T1,LY1,OT4,QX,T5},
6389   family = cmr }
6390 {

```



latex.ltx has:

```

\def\nonfrenchspacing{
  \sfcode`. 3000
6391   . = {333,2000,-667},
      \sfcode`? 3000
6392   ? = {333,2000,-667},
      \sfcode`! 3000
6393   ! = {333,2000,-667},
      \sfcode`: 2000
6394   : = {333,1000,-500},
      \sfcode`; 1500
6395   ; = {   , 500,-333},
      \sfcode`\, 1250
6396   {,}= {   , 250,-200}
      }
6397 }
6398
```

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.

```

6399 \SetExtraSpacing
6400 [ name    = nonfrench-default,
6401   load     = default,
6402   context  = nonfrench ]
6403 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6404 {
6405   . = {240,2000,-667},
6406   ? = {240,2000,-667},
6407   ! = {240,2000,-667},
6408   : = {240,1000,-500},
6409   ; = {   , 500,-333},
6410   {,}= {   , 250,-200}
6411 }
6412
```

## 15.10 Additional Kerning

Default unit is 1em.

```

6413 %%% -----
6414 %%% ADDITIONAL KERNING
6415
```

A dummy list to be loaded when no context is active.

```
6416 \SetExtraKerning
6417   [ name = empty ]
6418   { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
6419   { }
6420
```

### 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. 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.)

```
6421 \SetExtraKerning
6422   [ name      = french-default,
6423     context   = french,
6424     unit      = space ]
6425   { encoding = {OT1,T1,LY1} }
6426   {
6427     : = {1000,}, % = \fontdimen2
6428     ; = {500, }, % ~ \thinspace
6429     ! = {500, },
6430     ? = {500, }
6431   }
6432
```

This has the disadvantage that the word following a left guillemet will not be hyphenated. This might be fixed in pdfT<sub>E</sub>X.

```
6433 \SetExtraKerning
6434   [ name      = french-guillemets,
6435     context   = french-guillemets,
6436     load      = french-default,
6437     unit      = space ]
6438   { encoding = {T1,LY1} }
6439   {
6440     \guillemotleft = { ,800}, % = 0.8\fontdimen2
6441     \guillemotright = {800, }
6442   }
6443
6444 \SetExtraKerning
6445   [ name      = french-guillemets-OT1,
6446     context   = french-guillemets,
6447     load      = french-default,
6448     unit      = space ]
6449   { encoding = OT1 }
6450   { }
6451
```

### 15.10.2 Turkish

```
6452 \SetExtraKerning
6453   [ name      = turkish,
```

---

16 [http://fr.wikipedia.org/wiki/Espace\\_typographique](http://fr.wikipedia.org/wiki/Espace_typographique).

```

6454     context = turkish ]
6455     { encoding = {OT1,T1,LY1} }
6456     {
6457         : = {167, }, % = \thinspace
6458         ! = {167, },
6459         {=} = {167, }
6460     }
6461
6462 </m-t>
6463 </config>

```

## 16 Auxiliary File for Micro Fine Tuning

This file can be used to test protrusion and expansion settings.

```

6464 < *test>
6465 \documentclass{article}
6466
6467 %% Here you can specify the font you want to test, using
6468 %% the commands \fontfamily, \fontseries and \fontshape.
6469 %% Make sure to end all lines with a comment character!
6470 \newcommand*\TestFont{%
6471     \fontfamily{ppl}%
6472     %% \fontseries{b}%
6473     %% \fontshape{it}% sc, sl
6474 }
6475
6476 \usepackage{ifthen}
6477 \usepackage[T1]{fontenc}
6478 \usepackage[latin1]{inputenc}
6479 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
6480
6481 \pagestyle{empty}
6482 \setlength{\parindent}{0pt}
6483 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
6484 \newcommand*\testprotrusion[2][]{%
6485     \ifthenelse{\equal{#1}{r}}{\}{#2}%
6486     lorem ipsum dolor sit amet,
6487     \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
6488     \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6489     you know the rest%
6490     \ifthenelse{\equal{#1}{l}}{\}{#2}%
6491     \linebreak
6492     {\fontencoding{\encodingdefault}%
6493     \fontseries{\seriesdefault}%
6494     \fontshape{\shapedefault}%
6495     \selectfont
6496     Here is the beginning of a line, \dotfill and here is its end}\linebreak
6497 }
6498 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
6499 \def\stripprefix#1>{}
6500 \newcount\charcount
6501 \begin{document}
6502
6503 \microtypesetup{expansion=false}
6504
6505 {\centering The font in this document is called by:\\
6506 \texttt{\showTestFont}\par}\bigskip
6507
6508 \TestFont\selectfont
6509 This line intentionally left empty\linebreak
6510 %% A -- Z
6511 \charcount=65

```

```

6512 \loop
6513   \testprotrusion{\char\charcount}
6514   \advance\charcount 1
6515   \ifnum\charcount < 91 \repeat
6516 %% a -- z
6517 \charcount=97
6518 \loop
6519   \testprotrusion{\char\charcount}
6520   \advance\charcount 1
6521   \ifnum\charcount < 123 \repeat
6522 %% 0 -- 9
6523 \charcount=48
6524 \loop
6525   \testprotrusion{\char\charcount}
6526   \advance\charcount 1
6527   \ifnum\charcount < 58 \repeat
6528 %%
6529   \testprotrusion[r]{,}
6530   \testprotrusion[r]{.}
6531   \testprotrusion[r]{;}
6532   \testprotrusion[r]{:}
6533   \testprotrusion[r]{?}
6534   \testprotrusion[r]{!}
6535   \testprotrusion[l]{\textexclamdown}
6536   \testprotrusion[l]{\textquestiondown}
6537   \testprotrusion[r]{}}
6538   \testprotrusion[l]{({}
6539   \testprotrusion{/}
6540   \testprotrusion{\char`\}
6541   \testprotrusion{-}
6542   \testprotrusion{\textendash}
6543   \testprotrusion{\textemdash}
6544   \testprotrusion{\textquoteleft}
6545   \testprotrusion{\textquoteright}
6546   \testprotrusion{\textquotedblleft}
6547   \testprotrusion{\textquotedblright}
6548   \testprotrusion{\quotesinglbase}
6549   \testprotrusion{\quotedblbase}
6550   \testprotrusion{\guilsinglleft}
6551   \testprotrusion{\guilsinglright}
6552   \testprotrusion{\guillemotleft}
6553   \testprotrusion{\guillemotright}
6554
6555 \newpage
6556 The following displays the current font stretched by 5%,
6557 normal, and shrunk by 5%:
6558
6559 \bigskip
6560 \newlength{\MTln}
6561 \newcommand*\teststring
6562   {ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
6563 \settowidth{\MTln}{\teststring}
6564 \microtypesetup{expansion=true}
6565
6566 \parbox{1.05\MTln}{\teststring\linebreak\}
6567   \teststring\par\bigskip
6568 \parbox{0.95\MTln}{\teststring}
6569
6570 \end{document}
6571 </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> ) .....	74	<code>\MT@get@basefamily</code> : only remove suffix, if it is 'x' or 'j' .....	75
issue an error instead of a warning, when pdfTeX version is too old for autoexpand .....	115	<code>\MT@get@listname@</code> : don't check for empty attributes list .....	75
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i> ) .....	123	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i> ) .....	40
Protrusion: add factors for some more characters settings for Adobe Minion (contributed by <i>Harald Harders</i> ) .....	128	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i> ) .....	79
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance .....	129	<code>\MT@pdfTeX@no</code> : fix concerning version check (reported by <i>Harald Harders</i> ) .....	34
<code>\MT@declare@sets</code> : remove spaces around set name .....	88	<code>\MT@permute</code> : don't use sets for empty encoding ..	102
<code>\MT@DeclareSet</code> : remove spaces around first argument .....	88	<code>\MT@pr@split</code> : fix: allow zero and negative values ..	55
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded ..	74	<code>\MT@use@set</code> : remove spaces around set name .....	92
		<code>\UseMicrotypeSet</code> : remove spaces around first argument .....	92

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

Font Sets: declare cmor as an alias of cmr .....	121	changed .....	89
new: allmath and basicmath .....	120	<code>\MT@get@inh@list</code> : fix: set inheritance list <code>\globally</code> to <code>\empty</code> .....	77
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding .....	149	<code>\MT@get@listname@</code> : alternatively check for alias font name .....	75
add settings for Computer Modern Roman math symbols .....	154	<code>\MT@get@size</code> : additional magic to catch some errors hijack <code>\set@fontsize</code> instead of <code>\@setfontsize</code> ..	90
<code>\MT@encoding@check</code> : check whether only one encoding specified .....	101	<code>\MT@loop</code> : fix: new macro, used instead of <code>\loop</code> ..	44
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement .....	52	<code>\MT@maybe@do</code> : also check for alias font name .....	52
<code>\MT@get@basefamily</code> : also remove 'w' (swash capitals) .....	75	<code>\MT@permute@@@@@</code> : more sanity checks for <code>\SetProtrusion</code> and <code>\SetExpansion</code> .....	103
<code>\MT@get@highlevel</code> : check whether defaults have		<code>\MT@setupfont</code> : also search for alias font file .....	49
		fix: call <code>\@enc@update</code> if necessary .....	50

### Version 1.3 (2004/10/27)

General: fix: specifying load option does no longer require to give a name, too .....	98	<code>\MT@catcodes</code> : check some category codes (compatibility with german) .....	36
Font Sets: declare aer, zer and hfor as an alias of cmr .....	121	<code>\MT@load@list</code> : check whether list exists .....	73

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

General: check for pdfcprot .....	47	(OT1, T1, lmr) .....	134
don't use scratch registers in global definitions ..	78	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options .....	111
use <code>\pickup@font</code> instead of <code>\define@newfont</code> as the hook for <code>\MT@setupfont</code> .....	83	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too .....	95
use one instead of five counters .....	44		
Protrusion: tweak quote characters for cmr variants			

**Version 1.4a (2004/11/17)**

General: new option: <code>final</code> . . . . .	108	when reading files (reported by <i>Michael Hoppe</i> )	74
<code>\MT@normal@catcodes</code> : 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> ) . .	109	name if encoding failed . . . . .	53
new message if <code>\pdfoutput</code> is changed . . . . .	114	<code>\MT@get@basefamily</code> : fix: failed for font names of the form <code>abczz</code> (reported by <i>Georg Verwey</i> ) . . . .	75
optimisation: use less <code>\expandafters</code> and <code>\csnames</code>	38	<code>\MT@get@slot</code> : don't define <code>\MT@char</code> globally (save stack problem) . . . . .	78
Protrusion: harmonise dashes in upshape and italic ( <code>cmr</code> , <code>pad</code> , <code>ppl</code> ) . . . . .	128	<code>\MT@ifdimen</code> : don't set <code>\MT@count</code> globally (save stack problem) . . . . .	41
slanted like italics . . . . .	137	<code>\MT@use@set</code> : don't use undeclared font sets . . . . .	92
<code>\MT@checklist@family</code> : fix: don't try alias family			

**Version 1.5 (2004/12/15)**

General: defaults: step: 4 (suggested by <i>Hàn Thê Thành</i> ) . . . . .	108	<code>\MT@catcodes</code> : reset catcode of <code>^</code> (compatibility with <code>chemsym</code> ) . . . . .	36
defaults: calculate step as <code>min(stretch,shrink)/5</code>	114	<code>\MT@get@highlevel</code> : don't test defaults if called after begin document . . . . .	89
defaults: turn off expansion for DVI output . . .	114	<code>\MT@normal@catcodes</code> : reset catcode of <code>'</code> (compatibility with Turkish <code>babel</code> ) . . . . .	74
disable automatic expansion for DVI output . . .	115	<code>\MT@scale@factor</code> : warning for factors outside limits	57
new option: selected, by default false (suggested by <i>Hàn Thê Thành</i> ) . . . . .	107	<code>\MT@scale@to@em</code> : don't use <code>\lpcode</code> and <code>\rprcode</code> for the calculation . . . . .	56
Documentation: add 'Short History' . . . . .	28	<code>\MT@set@ex@codes</code> : allow non-selected font expansion . . . . .	61
add note about DVI output option . . . . .	8	<code>\MT@set@pr@codes</code> : adjust protrusion factors before setting the inheriting characters . . . . .	54
Inheritance: remove <code>\ss</code> from T1 list, add <code>\DJ</code> . .	123		
Protrusion: settings for Bitstream Charter . . . .	129		
<code>\DeclareMicrotypeAlias</code> : remove spaces around arguments . . . . .	93		

**Version 1.6 (2005/01/24)**

General: defaults: turn off expansion for old pdf $\TeX$ versions . . . . .	109	improve settings for numbers (pointed out by <i>Peter Muthesius</i> ) . . . . .	131
disable automatic expansion for old pdf $\TeX$ versions . . . . .	115	tune CMR math letters (OML encoding) . . . . .	154
load a font, if none is active . . . . .	49	<code>\MT@get@charwd</code> : use e- $\TeX$ 's <code>\fontcharwd</code> , if available	56
new option: factor, by default 1000 . . . . .	108	<code>\MT@get@inh@list</code> : correct message if selected is false . . . . .	77
restructure dtx file . . . . .	120	<code>\MT@set@ex@codes</code> : introduce factor option . . . . .	61
test whether <code>\pickup@font</code> has changed . . . . .	85	<code>\MT@set@pr@codes</code> : introduce factor option . . . . .	54
test whether numeric options receive a number	108	<code>\MT@use@set</code> : retain current set if new set is undeclared . . . . .	92
use e- $\TeX$ 's <code>\ifcsname</code> and <code>\ifdefined</code> if defined .	39	<code>\MT@vinfo</code> : new macro: used instead of <code>\ifMT@verbose</code> . . . . .	33
Protrusion: add italic uppercase Greek letters . .	137		

**Version 1.6a (2005/02/02)**

Documentation: add table of fonts with tailored protrusion settings . . . . .	20	reported by <i>Bernard Gaulle</i> ) . . . . .	78
<code>\MT@get@slot</code> : completely redone, hopefully more robust (compatible with <code>frenchpro</code> ; problem		<code>\MT@pdftex@no</code> : new macro . . . . .	34
		<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older pdf $\TeX$ versions . . . . .	62

**Version 1.7 (2005/03/23)**

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i> ) . . . . .	89	don't allow automatic expansion for old pdf $\TeX$ versions . . . . .	100
----------------------------------------------------------------------------------------------	----	-----------------------------------------------------------------------	-----

fix: remove space after autoexpand	100	<code>\MT@if@list@exists</code> : don't define <code>\MT@#1@c@name</code> globally, here and elsewhere	77
modify <code>\showhyphens</code>	116	<code>\MT@ifdimen</code> : comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i> )	41
new value for verbose option: errors	108	<code>\MT@increment</code> : use e-TeX's <code>\numexpr</code> if available	44
shorter command names	45	<code>\MT@is@composite</code> : new macro: construct command for composite character; no uncontrolled expansion	81
warning when running in draft mode	113	<code>\MT@normal@catcodes</code> : reset catcode of <code>'</code> (compatibility with french* packages)	74
Documentation: add hint about compatibility	25	<code>\MT@scale</code> : new macro: use e-TeX's <code>\numexpr</code> if available	44
remove table of match order	12	<code>\MT@set@ex@codes</code> : two versions of this macro	61
Protrusion: fix: remove <code>\</code> from OT1, add <code>\textbackslash</code> to T1 encoding	132	<code>\MT@split@name</code> : don't define <code>\MT@encoding</code> &c. globally	52
<code>\DeclareMicrotypeAlias</code> : may also be used inside configuration files	93	<code>\MT@test@ast</code> : make it simpler	89
<code>\LoadMicrotypeFile</code> : new command (suggested by <i>Andreas Böhmann</i> )	94	<code>\MT@try@order</code> : always check for size, too (suggested by <i>Andreas Böhmann</i> )	76
<code>\MicrotypeHook</code> : new command for font package authors	110	fix: also check for <code>//&lt;series&gt;/&lt;shape&gt;/</code> (reported by <i>Andreas Böhmann</i> )	76
<code>\microtypesetup</code> : fix: warning also when setting to (no)compatibility	111	<code>\MT@warn@code@too@large</code> : new macro: type out maximum protrusion factor	58
<code>\MT@begin@catcodes</code> : also use inside configuration commands	75	<code>\MT@warn@err</code> : new macro: for verbose=errors	33
<code>\MT@get@listname@</code> : use <code>\@tfor</code> ( <i>Andreas Böhmann's</i> idea)	75		
<code>\MT@get@slot</code> : remove backslash hack	78		
test for <code>\chardef</code> commands	79		
test whether <code>\&lt;encoding&gt;\&lt;...&gt;</code> is defined	78		

## Version 1.8 (2005/06/23)

General: <code>\SetProtrusion</code> : new key: unit	99	<code>\MT@get@charwd</code> : warning for missing (resp. zero- width) characters	56
if font substitution has occurred, set up the substi- tute font, not the selected one	83	<code>\MT@get@font@dimen@six</code> : new macro: test whether <code>\fontdimen 6</code> is defined	55
new option: config to load a different main config- uration file	109	<code>\MT@get@listname@</code> : made recursive	75
new option: unit, by default character	109	<code>\MT@get@slot</code> : fix: expand active characters	78
Documentation: add example for factor option	13	test whether <code>\&lt;encoding&gt;\&lt;...&gt;</code> is defined made more robust	78
add example of how to get rid of a widow (sugges- ted by <i>Adam Kucharczyk</i> )	14	<code>\MT@get@unit</code> : new macro: get unit for codes	59
add hint about error messages	25	<code>\MT@in@rlist</code> : made recursive	43
Font Sets: add U encoding to allmath	120	<code>\MT@is@active</code> : new macro: translate inputenc- defined characters	80
declare <code>pxr</code> and <code>txr</code> as aliases of <code>ppl</code> resp. <code>ptm</code>	121	<code>\MT@is@letter</code> : warning for non-ASCII characters	79
Inheritance: remove <code>\DJ</code> from T1 list (it's the same as <code>\DH</code> )	123	<code>\MT@led@kern</code> : character protrusion with <code>ledmac</code>	46
Protrusion: add LY1 characters for Times	136	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\@for</code>	42
settings for AMS math fonts	158	<code>\MT@map@tlist@n</code> : new macro: used instead of <code>\@tfor</code>	43
verified settings for slanted Computer Modern Ro- man	143	<code>\MT@normal@catcodes</code> : reset catcodes of the remain- ing ASCII characters	74
<code>\add@accent</code> : fix: disable micro-typographic setup in- side <code>\add@accent</code> (reported by <i>Stephan Hennig</i> )	84	<code>\MT@old@cmd</code> : renamed commands from <code>\..MicroType..</code> to <code>\..Microtype..</code>	32
<code>\DeclareMicrotypeAlias</code> : warning when overriding an alias font	93	<code>\MT@pdftex@no</code> : case 5: pdfTeX 1.30	34
<code>\DeclareMicrotypeSetDefault</code> : new command: set default font set	92	<code>\MT@permute@@@@@</code> : add ranges to the beginning of the lists	103
<code>\MT@check@rlist</code> : made recursive	104	<code>\MT@pr@split</code> : get character width once only	55
<code>\MT@curr@list@name</code> : new macro: current list type and name	82	<code>\MT@scale</code> : fix: remove spaces in e-TeX variant (repor- ted by <i>Mark Rossi</i> )	44
<code>\MT@declare@sets</code> : warning when redefining a set	88	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when hyperref is loaded	47
<code>\MT@define@set@key@</code> : use comma lists instead of token lists	88	restore csquotes's active characters	47
<code>\MT@find@file</code> : no longer wrap names in commands	74	restore percent character if Spanish babel is loaded	47
		<code>\MT@use@set</code> : fix: remove braces in first line	92
		<code>\MT@xadd</code> : simplified	42

**Version 1.9 (2005/10/28)**

General: <code>\DeclareMicrotypeSet</code> : new key: font ..	91	settings for OT4 encoding (Computer Modern Roman, Palatino, Times) .....	128
<code>\SetProtrusion</code> : value ‘relative’ renamed to ‘character’ for key unit .....	99	settings for T5 encoded Computer Modern Roman .....	128
allow context-specific font setup .....	83	<code>\DisableLigatures</code> : new command: disable ligatures (requires pdf $\TeX$ 1.30) .....	94
disable expansion if both step and shrink are zero .....	116	<code>\microtypecontext</code> : new command: change setup context in the document .....	86
disable microtype setup inside hyperref’s <code>\pdfstringdef</code> (reported by <i>Hàn Thê Thành</i> ) ..	48	<code>\MT@checklist@family</code> : fix: add two missing <code>\expandafters</code> .....	53
fix: use true as the default value .....	106	<code>\MT@detokenize@c</code> : fix the $\mathcal{E}$ - $\TeX$ version .....	40
option unit: rename value relative to character .....	109	<code>\MT@exp@two@n</code> : new macros: less <code>\expandafters</code> ..	39
warning if user requested zero step .....	114	<code>\MT@get@opt</code> : new key ‘preset’ to set all characters to the specified value before loading the lists ..	59
Documentation: add hint about verbatim environment .....	25	<code>\MT@is@active</code> : redone: use <code>\set@display@protect</code> ..	80
add remark about Type 1 fonts required for automatic font expansion .....	7	<code>\MT@is@letter</code> : using <code>\catcode</code> should be more efficient than inspecting the <code>\meaning</code> .....	79
Font Sets: add OT4 encoding to text sets .....	120	<code>\MT@maybe@do</code> : redone .....	52
add T5 encoding to text sets .....	120	<code>\MT@pdf@tex@no</code> : compatibility with $\TeX$ Live hack (reported by <i>Herbert Voß</i> ) .....	35
declare <code>qpl</code> and <code>qtm</code> (qfonts, $\TeX$ Gyre) as aliases of <code>ppl</code> resp. <code>ptm</code> .....	121	<code>\MT@rem@from@clist</code> : new macro: remove an item from a comma list .....	43
Inheritance: add list for OT4 .....	125	<code>\MT@scale@factor</code> : generalised .....	57
add list for T5 (requested by <i>Hàn Thê Thành</i> ) ..	126	<code>\MT@toks</code> : use instead of <code>\toks@</code> .....	36
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR .....	132	<code>\SetProtrusion</code> : (et al.) new key: font .....	95

**Version 1.9a (2005/12/05)**

General: ‘ <code>(file name)/(line number)</code> ’ as default list name .....	98	diately (requested by <i>Georg Verwey</i> ) .....	88
new option: <code>defersetup</code> , by default true .....	107	<code>\MT@get@highlevel</code> : no longer check whether defaults have changed .....	89
remove superfluous test whether <code>\pickup@font</code> has changed .....	85	<code>\MT@ifdefined@c@T</code> : new macros: true case only ..	39
Documentation: add explanation for error message in DVI mode .....	25	<code>\MT@ifint</code> : use <code>\pdfmatch</code> if available .....	40
add explanation for error message with non-Type 1 fonts .....	26	<code>\MT@ifstreq</code> : use <code>\pdfstrcmp</code> if available .....	41
Font Sets: declare <code>mbch</code> (mathdesign) as an alias of Charter .....	122	<code>\MT@in@clist</code> : fix .....	43
Protrusion: fix: remove <code>_</code> from OT1 encoding ..	133	<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) .....	57
settings for T5 encoded Charter .....	128	<code>\MT@is@feature</code> : new macro: check for pdf $\TeX$ feature .....	45
<code>\microtypesetup</code> : inside the preamble, accepts all package options .....	111	<code>\MT@map@clist@n</code> : following $\LaTeX$ 3 .....	42
<code>\MT@check@font@cx</code> : optimise context-sensitive setup ..	85	<code>\MT@permute@@@@@</code> : don’t define permutations for unused encodings .....	103
<code>\MT@define@set@key@</code> : don’t expand variables immediately (requested by <i>Georg Verwey</i> ) .....	88	<code>\MT@rem@from@clist</code> : fix .....	43
<code>\MT@get@highlevel</code> : no longer check whether defaults have changed .....	89	<code>\MT@setup@</code> : defer setup until the end of the preamble ..	46
<code>\MT@ifdefined@c@T</code> : new macros: true case only ..	39		
<code>\MT@ifint</code> : use <code>\pdfmatch</code> if available .....	40		
<code>\MT@ifstreq</code> : use <code>\pdfstrcmp</code> if available .....	41		
<code>\MT@in@clist</code> : fix .....	43		
<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) .....	57		
<code>\MT@is@feature</code> : new macro: check for pdf $\TeX$ feature .....	45		
<code>\MT@map@clist@n</code> : following $\LaTeX$ 3 .....	42		
<code>\MT@permute@@@@@</code> : don’t define permutations for unused encodings .....	103		
<code>\MT@rem@from@clist</code> : fix .....	43		
<code>\MT@setup@</code> : defer setup until the end of the preamble ..	46		

**Version 1.9b (2006/01/20)**

General: compatibility with <code>listings</code> : sanitise more catcodes (reported by <i>Holger Uhr</i> ) .....	48	add samples of micro-typographic features .....	4
compatibility with the <code>extendedchar</code> option of the <code>listings</code> package .....	48	<code>\MT@features</code> : use throughout the package to adjust to beta-ness .....	45
Documentation: activate expansion in the distributed PDF .....	1	<code>\MT@ifdimen</code> : use <code>\pdfmatch</code> if available .....	41
		<code>\MT@warn@code@too@large</code> : fix calculation with present factor .....	58



**Version 1.9c (2006/02/02)**

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verwey</i> )	21	<code>\MT@define@code@key@font</code> : fix: context was ignored	97
Protrusion: settings for URW Garamond	129	<code>\MT@define@code@key@size</code> : fix: embrace <code>\MT@tempsize</code> in <code>\csname</code> (bug introduced in 1.9b)	97

**Version 1.9d (2006/05/05)**

Font Sets: <code>md*</code> instead of <code>m</code> series in basic sets	120	<code>\MT@get@font@dimen</code> : warning for zero <code>fontdimen</code>	57
add QX encoding to text sets	120	<code>\MT@get@opt</code> : optimise: don't reset when preset option is set	59
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i> )	125	set list name before presetting	59
Protrusion: settings for QX encoding (contributed by <i>Maciej Eder</i> )	134	<code>\MT@is@active</code> : support for Unicode (inputenc/utf8)	80
settings for Euro symbols (Adobe, ITC, marvosym)	166	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>tex4ht</code> is loaded (reported by <i>Peter Dybala</i> )	47
tweak AMS settings	158	<code>\SetProtrusion</code> : (et al.) optimise: unify keys for mandatory argument	95
<code>\DeclareCharacterInheritance</code> : fix: empty context	100	(et al.) split keys of optional and mandatory argument	95
<code>\MT@detokenize@n</code> : new macro: use <code>\detokenize</code> if available	40		
<code>\MT@get@ex@opt</code> : fix: evaluate preset	63		

**Version 1.9e (2006/07/28)**

General: fix: default value for <code>activate</code> : true	106	settings for Euler Roman font	161
Documentation: add hint about unknown encodings	24	<code>\DeclareCharacterInheritance</code> : new key 'inputenc' to set the input encoding	100
include LPPL	186	<code>\MT@rem@from@clist</code> : model after <code>\@removeelement</code>	43
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> )	122	<code>\MT@setup@</code> : empty <code>\MT@setup@</code> after use (compatibility with the <code>combine</code> class)	46
Inheritance: adapt to <code>marvosym</code> 's changed encoding	127	<code>\pickup@font</code> : no tracing with <code>trace</code> package	84
Protrusion: complete settings for Euler Fraktur and Script fonts	165	<code>\SetExpansion</code> : new key: <code>inputenc</code>	95
fix: forgotten comma in <code>mt-mvs.cfg</code> ; adapt to <code>marvosym</code> 's changed encoding	166	<code>\SetProtrusion</code> : (et al.) new key: <code>inputenc</code>	95

**Version 1.9f (2006/09/09)**

Protrusion: fix: <code>euler-vm</code> did not load <code>euler</code> settings	162	<code>\MT@reset@context</code> : only reset context if it has actually been changed	86
<code>\MT@curr@list@name</code> : fix: <code>\MessageBreak</code> must not be expanded	82	<code>\MT@set@inh@list</code> : fix: forgotten comma in the features list	101
<code>\MT@gdef@n</code> : new macros: global variants	39	<code>\MT@set@named@keys</code> : new macro: set name first, simplify parsing of optional argument	97
<code>\MT@get@inh@list</code> : fix: input encoding must be set after the inheritance list has been parsed	78	<code>\SetProtrusion</code> : (et al.) set catcodes before parsing optional argument	95
<code>\MT@glet</code> : new macro	38		

**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> )	107	<i>Daniel Flipo</i> )	119
(beta:1) new option: <code>letterspace</code> , by default 100	108	compatibility with <code>listings</code> : set catcode of backslash to zero (reported by <i>Steven Bath</i> )	48
(beta:3) compatibility with <code>soul</code> : register <code>\textls</code> and <code>\sstyle</code>	49	maybe disable <code>\MT@noligatures</code> after the preamble	118
(beta:8) option 'babel': disable Turkish <code>babel</code> 's active characters	119	new package <code>letterspace</code> : a stripped-down version, containing the <code>letterspacing</code> commands only	1
(beta:8) option 'babel': fix: switch off French <code>babel</code> 's active characters with <code>\shorthandoff</code> instead of <code>\NoAutoSpaceBeforeFDP</code> (reported by		option 'unit', <code>\SetProtrusion</code> : deprecate value 'relative' completely	99
		option 'letterspacing' renamed to 'letterspace'	108
		Documentation: add hint about how to increase	

font_max and font_mem_size .....	26	totally redone, using the new \letterspacefont .....	69
add hint about warning when tracking <i>and</i> expansion is applied to a font .....	26	\MT@declare@sets: fix: empty size list when redefining set .....	88
add remark about ‘draft’ option disabling microtype (noted by <i>Michalis Miatidis</i> ) .....	9	\MT@is@symbol: made even more robust .....	80
qualify hint about web documents with regard to older pdfTeX versions .....	24	\MT@load@inputenc: sanitise catcodes before loading input encoding (problem with listings) .....	60
qualify hints about expansion error messages with regard to older pdfTeX versions .....	26	\MT@pdfTeX@no: (beta:1) case 6: pdfTeX 1.40 .....	34
Font Sets: add footnotesize and scriptsize sets .....	120	\MT@split@name: adjust to possible letterspacing ..	52
add smallcaps set .....	120	\SetExtraKerning: (beta:1) new command: additional kerning .....	96
Kerning: remove letterspacing settings .....	171	\SetExtraSpacing: (beta:1) new command: adjustment of interword spacing .....	96
\DeclareMicrotypeBabelHook: (beta:1) new command: interaction with babel .....	94	\SetTracking: new command: tracking .....	96
\lssstyle: (beta:1) new command: letterspacing ..	69	\textls: (beta:1) new command: letterspacing ...	72
(beta:8) fix: font switches don’t pose a problem anymore .....	69	starred version: remove spaces around text ....	72
(beta:8) fix: letterspacing commands may be nested .....	69	\tracingmicrotypeinpdf: new debug method: mark all fonts with PDF annotations .....	33

### Version 2.1 (2007/01/21)

General: compatibility with CJK: also check for its definition .....	83	\lslig: new command: protect ligatures in letter-spaced text .....	69
compatibility with pinyin: disable microtype in \py@macron (reported by <i>Sven Naumann</i> ) ....	49	\MT@get@ls@basefont: redone: use \pdfmatch to make it bullet-proof .....	69
fix: letterspace package forgot to load keyval ..	36	\textls: fix: use \hmode@bgroup .....	72

### Version 2.2 (2007/07/14)

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i> ) .....	46	\MT@get@charwd: subtract letterspacing amount from width .....	57
new option: copyfonts .....	108	\MT@get@ls@basefont: fix again: remember base font in a macro .....	69
simplify key declarations .....	98	\MT@ifdimen: employ LuaTeX features if available ..	41
warning if stretch or shrink aren’t multiples of step .....	116	\MT@ifint: employ LuaTeX features if available ...	40
Documentation: add hint about error message with pdfTeX 1.40 .....	25	\MT@ifstreq: employ LuaTeX features if available ..	41
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i> ) .....	24	fix: $\TeX$ version shouldn’t use \x and \y (found by <i>Wiebke Petersen</i> ) .....	41
add overview .....	6	\MT@is@composite: more robust: expand exactly once	81
logo transparency and amusement .....	1	\MT@is@symbol: expand once more (for frenchpro)	81
Font Sets: declare FPL Neu as an alias of Palatino ..	121	\MT@lsfont: use \font@name, not \MT@font .....	68
declare chr (chmath) as an alias of Charter (reported by <i>Geoff Vallis</i> ) .....	122	\MT@maybe@etex: use catcode trickery .....	38
default set for tracking: smallcaps .....	121	\MT@pdfTeX@no: case 7: pdfTeX 1.40.4 .....	34
Inheritance: remove ‘-’ $\rightarrow$ ‘127’ .....	124	\MT@preset@aux@space: generalised .....	60
Protrusion: settings for Bitstream Letter Gothic ..	129	\MT@requires@luatex: (basic) support for LuaTeX ..	38
Spacing: add sample .....	167	\MT@set@all@pr: (et al.) allow empty values .....	54
Tracking: add ligatures that are to be disabled ...	127	\MT@set@inputenc: only load inputenc files if necessary .....	59
\DeclareMicrotypeVariants: new command .....	93	\MT@set@tr@codes: disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	68
\DisableLigatures: new optional argument: disable selected ligatures only .....	94	possibility to customise interword spacing ....	68
\lslig: always defined .....	69	\MT@setupfont: don’t call \@@enc@update anymore ..	50
\MT@checklist@font: fix: construct font name from characteristics .....	54	only add features that are available with the respective pdfTeX .....	50
\MT@compensate@tr: adjust protrusion of letterspaced fonts .....	55	\MT@setupfont@hook: restore percent character if Galician babel is loaded .....	47
\MT@copy@font: optionally work on copies of fonts ..	50	\MT@tracking: remember fonts that shouldn’t be letterspaced .....	67
\MT@get@basefamily: redone, working on font names and suffixes of arbitrary length .....	75	\MT@tracking@: fix: tracking couldn’t be re-enabled	67

<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	117	new keys ‘spacing’ and ‘outer spacing’ to adjust interword spacing (suggested by <i>Steven E. Harris</i> )	96
<code>\MT@with@babel@and@T</code> : also inspect class options	46	third argument may be empty	96
<code>\pickup@font</code> : <code>letterspace.sty</code> : setup inside group	84	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code>	86
<code>\SetTracking</code> : new key ‘no ligatures’ to disable ligatures of letterspaced fonts	96		

## B Index

Links refer to the page where the corresponding entry is described (bold face) resp. occurs. Plain numbers refer to the code line where the corresponding entry is defined (underlined>) resp. used.

### Options:

<code>DVIoutput</code>	8
<code>activate</code>	6
<code>auto</code>	7
<code>babel</code>	9
<code>config</code>	9
<code>draft</code>	9
<code>expansion</code>	6
<code>factor</code>	7
<code>final</code>	9
<code>kerning</code>	7
<code>letterspace</code>	8
<code>protrusion</code>	6
<code>selected</code>	8
<code>shrink</code>	8
<code>spacing</code>	7
<code>step</code>	8
<code>stretch</code>	8
<code>tracking</code>	7
<code>unit</code>	7
<code>verbose</code>	9

### User Commands:

<code>\DeclareCharacterInheritance</code>	19
<code>\DeclareMicrotypeAlias</code>	19
<code>\DeclareMicrotypeBabelHook</code>	22
<code>\DeclareMicrotypeSet</code>	10
<code>\DeclareMicrotypeSet*</code>	10
<code>\DeclareMicrotypeSetDefault</code>	12
<code>\DeclareMicrotypeVariants</code>	19
<code>\DeclareMicrotypeVariants*</code>	19
<code>\DisableLigatures</code>	23
<code>\LoadMicrotypeFile</code>	20
<code>\lslig</code>	22
<code>\lsstyle</code>	22
<code>\microtypecontext</code>	21
<code>\microtypesetup</code>	9
<code>\SetExpansion</code>	14
<code>\SetExtraKerning</code>	18
<code>\SetExtraSpacing</code>	17
<code>\SetProtrusion</code>	12
<code>\SetTracking</code>	16
<code>\textls</code>	22
<code>\textls*</code>	22
<code>\textmicrotypecontext</code>	21
<code>\UseMicrotypeSet</code>	11

### A

<code>aoposter</code> (package)	90
<code>activate</code> (option)	6, 106, 177
<code>\add@accent</code>	2206
<code>ae</code> (package)	20, 121
<code>amssymb</code> (package)	158
<code>article</code> (package)	21
<code>auto</code> (option)	7, 107

### B

<code>babel</code> (option)	9, 22, 28, 107, 177
<code>babel</code> (package)	3, 5, 17, 22, 23, 47, 74, 94, 119, 122, 174, 175, 177, 178

### C

<code>\cf@encoding</code>	757
<code>chemsym</code> (package)	174
<code>chmath</code> (package)	20, 122, 178
<code>CJK</code> (package)	49, 84, 178
<code>cm-super</code> (package)	7
<code>color</code> (package)	9, 114
<code>combine</code> (package)	46, 177
<code>config</code> (option)	9, 19, 29, 109, 175
<code>contour</code> (package)	114
<code>copyfonts</code> (option)	50, 107, 108, 178
<code>crop</code> (package)	114
<code>csquotes</code> (package)	27, 29, 47, 132, 175
<code>\curr@fontshape</code>	2160, 2168, 2170

### D

<code>\DeclareCharacterInheritance</code>	12, 19, 2895
<code>\DeclareMicrotypeAlias</code>	14, 19, 41, 2586
<code>\DeclareMicrotypeBabelHook</code>	16, 22, 35, 2639
<code>\DeclareMicrotypeSet</code>	3, 10, 28, 30, 42, 2314
<code>\DeclareMicrotypeSet*</code>	10, 2314
<code>\DeclareMicrotypeSetDefault</code>	5, 12, 32, 2545
<code>\DeclareMicrotypeVariants</code>	13, 19, 29, 34, 2573
<code>\DeclareMicrotypeVariants*</code>	19
<code>deferssetup</code> (option)	47, 107, 111, 176
<code>\define@newfont</code>	2167, 2169, 2172
<code>\DisableLigatures</code>	11, 23, 33, 2619
<code>\do@subst@correction</code>	2156
<code>docstrip</code> (package)	31
<code>draft</code> (option)	9, 107, 108, 178
<code>dsfont</code> (package)	55
<code>DVIoutput</code> (option)	8, 9, 29, 107, 174

### E

<code>\@enc@update</code>	757	<code>eco</code> (package)	20, 121
---------------------------	-----	----------------------------	---------

- `\efcode` ..... 1255, 1289, 1290, 1334, 1336  
`eucal` (package) ..... 163  
`eufrak` (package) ..... 165  
`euler` (package) ..... 82, 161, 177  
`eulervm` (package) ..... 20, 122, 162, 177  
`euroitc` (package) ..... 166  
`europs` (package) ..... 166  
`eurosans` (package) ..... 166  
`expansion` (option) ..... 6, 7, 11, 106, 114
- F**
- `\f@encoding` ..... 757  
`\f@size` ..... 2160, 2168, 2170  
`factor` (option) ..... 7, 13, 30, 108, 109, 174  
`fancyvrb` (package) ..... 25  
`final` (option) ..... 9, 30, 107, 108, 174  
`fix-cm` (package) ..... 51  
`\font` ..... 746  
`\font@name` ..... 83, 785, 787, 789, 791, 803,  
     804, 1574, 1579, 1607, 1619, 1624, 1626,  
     1629, 2167, 2169, 2172, 2192, 2194, 2216, 2236  
`\fontcharwd` ..... 1016  
`\fontdimen` 958, 965, 1038, 1043, 1144, 1147, 1590,  
     1591, 1614, 1660, 1664, 1671, 1681, 1685, 1688  
`fontinst` (package) ..... 169  
`fontinstallationguide` (package) ..... 147  
`french` (package) ..... 175  
`frenchpro` (package) ..... 81, 174, 178
- G**
- `german` (package) ..... 36, 173  
`graphics` (package) ..... 9, 114
- H**
- `hfoldsty` (package) ..... 20, 121  
`hyperref` (package) ..... 9, 47, 48, 83, 114, 175, 176
- I**
- `IEEEtran` (package) ..... 83  
`\iffontchar` ..... 1049  
`\ifMT@auto` ..... 192, 3538, 3583  
`\ifMT@babel` ..... 192, 3698, 3745  
`\ifMT@do` ..... 823, 884, 1532, 1778  
`\ifMT@document` ..... 234, 3013  
`\ifMT@draft` ..... 192, 3472  
`\ifMT@expansion` ..... 192, 3502, 3580  
`\ifMT@if@` ..... 191, 2230, 3117, 3765  
`\ifMT@inannot` ..... 72  
`\ifMT@inlist@` ..... 496, 541, 600,  
     622, 763, 851, 868, 877, 895, 913, 1530,  
     1829, 1840, 1888, 1936, 2224, 2288, 2300, 2604  
`\ifMT@kerning` ..... 192, 3643, 3687, 3758  
`\ifMT@noligatures` ..... 192, 3714  
`\ifMT@nonselected` ..... 1233, 1949, 1957  
`\ifMT@noreset` ..... 2011, 2018  
`\ifMT@opt@auto` ..... 3124, 3541  
`\ifMT@opt@DVI` ..... 3124, 3488  
`\ifMT@opt@expansion` ..... 3124, 3483  
`\ifMT@protrusion` ..... 192, 628, 650, 3490  
`\ifMT@selected` ..... 192, 3566, 3586  
`\ifMT@spacing` ..... 192, 3634, 3681, 3697  
`\ifMT@tracking` ..... 192, 3619, 3675  
`ifpdf` (package) ..... 9, 114
- `\iftracingmicrotypeinpdfall` ..... 77  
`inputenc` (package) ..... 14,  
     25, 29, 48, 59, 60, 78, 80, 82, 175, 177, 178
- J**
- `jurabib` (package) ..... 83
- K**
- `kerning` (option) ..... 7, 11, 28, 106  
`keyval` (package) ..... 36, 55, 122, 178  
`\knaccode` ..... 1483, 1484, 1496, 1499, 1505  
`\knbccode` ..... 1476, 1477, 1495, 1498, 1504  
`\knbscode` ..... 1379, 1380, 1405, 1409, 1415
- L**
- `ledmac` (package) ..... 27, 29, 46, 83, 175  
`ledpar` (package) ..... 27  
`\leftmarginkern` ..... 636  
`letterspace` (option) 8, 17, 22, 23, 28, 40, 67, 108, 177  
`letterspace` (package) 1, 23, 28, 31, 46, 68, 69, 177, 178  
`\letterspacefont` ..... 1579  
`lineno` (package) ..... 46  
`listings` (package) ..... 25, 48, 60, 75, 176–178  
`lmodern` (package) ..... 7, 134  
`\LoadMicrotypeFile` ..... 15, 20, 44, 2600, 3477  
`\lcode` ..... 940, 951, 989, 990, 1181, 1184  
`\lslig` ..... 22, 25, 1617  
`\lststyle` 22, 22, 704, 725, 1608, 1732, 1736, 1750, 3480
- M**
- `marvosym` (package) ..... 20, 28, 31, 127, 166, 177  
`mathdesign` (package) ..... 20, 122, 176  
`memoir` (package) ..... 21, 83  
`\Microtype@Hook` ..... 110, 3336  
`\microtypecontext` .....  
     ..... 18, 21, 2258, 2268, 3479, 3728, 3731  
`\microtypesetup` ..... 9, 17, 3340, 3478  
`minimal` (package) ..... 49  
`\MT@font` ..... 290, 766, 783, 914, 917,  
     960, 1039, 1053, 1542, 1600, 1891, 1962, 2195  
`\MT@abbr@ex` ..... 576  
`\MT@abbr@ex@c` ..... 576  
`\MT@abbr@ex@inh` ..... 576  
`\MT@abbr@kn` ..... 576  
`\MT@abbr@kn@c` ..... 576  
`\MT@abbr@kn@inh` ..... 576  
`\MT@abbr@nl` ..... 576  
`\MT@abbr@pr` ..... 576  
`\MT@abbr@pr@c` ..... 576  
`\MT@abbr@pr@inh` ..... 576  
`\MT@abbr@sp` ..... 576  
`\MT@abbr@sp@c` ..... 576  
`\MT@abbr@sp@inh` ..... 576  
`\MT@abbr@tr` ..... 576  
`\MT@abbr@tr@c` ..... 576  
`\MT@active@features` .....  
     793, 2218, 2221, 2233, 2240, 2277, 2287,  
     2626, 3491, 3581, 3626, 3635, 3644, 3720, 3721  
`\MT@addto@annot` ..... 72  
`\MT@addto@setup` ..... 614,  
     669, 1157, 2157, 2258, 2259, 2808, 3226,  
     3341, 3471, 3617, 3674, 3713, 3719, 3744, 3775  
`\MT@adjust@space@` ..... 1652–1654, 1657

- \MT@auto ..... 1242, 3537, 3550, 3559, 3563  
 \MT@auto@ ..... 1242, 1251, 1307  
 \MT@autofalse ..... 194, 3548, 3558  
 \MT@autotruer ..... 194, 3300  
 \MT@babelfalse ..... 201  
 \MT@babeltrue ..... 201  
 \MT@begin@catcodes 1830, 1831, 1869, 2608, 2621,  
     2646, 2660, 2682, 2696, 2710, 2900, 3313, 3314  
 \MT@cat ..... 1153, 2150, 2152  
 \MT@catcodes ..... 165, 185  
 \MT@char ..... 979, 989, 990, 995,  
     996, 999, 1001, 1016, 1017, 1049, 1181–  
     1185, 1277, 1289, 1290, 1292, 1294, 1334–  
     1336, 1368, 1379, 1380, 1386, 1387, 1393,  
     1394, 1397, 1399, 1405–1410, 1465, 1476,  
     1477, 1483, 1484, 1487, 1489, 1495–1499,  
     1787, 1788, 1991, 2007, 2008, 2013, 2082,  
     2085, 2087, 2113, 2116, 2960, 2961, 2966, 2969  
 \MT@char@ .....  
     1991, 1997, 2002, 2007, 2021, 2023, 2029,  
     2030, 2032, 2045, 2046, 2049, 2050, 2053,  
     2054, 2058, 2060, 2086, 2099, 2102, 2105, 2124  
 \MT@charstring ..... 2004, 2085, 2090  
 \MT@check@active@set .....  
     ..... 3328, 3497, 3601, 3628, 3638, 3648  
 \MT@check@font ..... 762, 2215, 2279  
 \MT@check@font@cx ..... 2219, 2279  
 \MT@check@range ..... 3088, 3090  
 \MT@check@range@ ..... 3090, 3091  
 \MT@check@rlist ..... 3041, 3081  
 \MT@check@rlist@ ..... 3081, 3082  
 \MT@check@step ..... 3589  
 \MT@checklist@ ..... 831, 844, 1775  
 \MT@checklist@family ..... 862  
 \MT@checklist@font ..... 906  
 \MT@checklist@size ..... 890  
 \MT@checksetup ..... 3350, 3361, 3368, 3388, 3407  
 \MT@clear@options ..... 141, 148, 276, 611, 3467  
 \MT@clist@break ..... 468, 857, 885, 901, 915  
 \MT@clist@function ..... 468  
 \MT@cnt@encoding ..... 2979, 2987, 2988  
 \MT@cnt@family ..... 2985, 2994, 2995  
 \MT@cnt@series ..... 2992, 3001, 3002  
 \MT@cnt@shape ..... 2999, 3007, 3008  
 \MT@compensate@tr ..... 946, 1568  
 \MT@compensate@tr@ ..... 946, 1568  
 \MT@compensate@tr@@ ..... 951, 953  
 \MT@config@file 3303, 3311, 3312, 3316, 3317, 3320  
 \MT@context ..... 1912, 1924, 1943, 1970  
 \MT@copy@font 751, 780, 2805, 2810, 3233, 3235, 3587  
 \MT@copy@font@ ..... 780, 2805, 2810, 3233, 3587  
 \MT@count ..... 406, 407, 558, 1009, 1016, 1018,  
     1019, 1024, 1029, 1030, 1034, 1043, 1076, 1138  
 \MT@curr@file ..... 1834, 1835,  
     1844, 1845, 2610, 2611, 2788, 2922, 3316, 3777  
 \MT@curr@list@name ..... 1041, 1079, 1169,  
     1303, 1815, 1981, 2119, 2127, 2132, 2140, 2146  
 \MT@curr@set@name .....  
     ..... 2339, 2341–2344, 2346, 2347, 2352,  
     2357, 2361, 2362, 2387, 2391, 2420, 2451,  
     2457, 2462, 2466, 2467, 2750, 2767, 2783,  
     2795, 2832, 2836, 2841, 2844, 2846, 2850,  
     2853, 2856, 2866, 2879, 2885, 2893, 2923, 2925  
 \MT@declare@char@inh ..... 2905, 2910, 2920  
 \MT@declare@sets ..... 2322, 2328, 2340, 2628  
 \MT@DeclareSet ..... 2317, 2319, 2336  
 \MT@DeclareSetAndUseIt ..... 2316, 2335  
 \MT@DeclareVariants ..... 2576, 2577, 2579  
 \MT@def@bool@opt .....  
     ... 3197, 3210–3212, 3221, 3231, 3239, 3247  
 \MT@def@n ..... 281, 2310, 2311  
 \MT@default@ex@set ..... 2561  
 \MT@default@kn@set ..... 2561  
 \MT@default@pr@set ..... 2561  
 \MT@default@sp@set ..... 2561  
 \MT@default@tr@set ..... 2561  
 \MT@define@code@key . 2732, 2775–2778, 2945–2947  
 \MT@define@code@key@font ..... 2755, 2780, 2949  
 \MT@define@code@key@size ..... 2743, 2779, 2948  
 \MT@define@opt@key .. 2781, 2797–2800, 2889, 2890  
 \MT@define@optionX ..... 3340, 3397, 3398  
 \MT@define@optionX@ ..... 3399, 3426, 3428, 3429  
 \MT@define@set@key@ ..... 2350, 2508–2511  
 \MT@define@set@key@font ..... 2455, 2513  
 \MT@define@set@key@size ..... 2380, 2512  
 \MT@detokenize@c ..... 324, 1999, 2083  
 \MT@detokenize@n ..... 324, 2115  
 \MT@dimen@six .. 957, 1010, 1034, 1076, 1680, 1683  
 \MT@dinfo ..... 60  
 \MT@dinfo@nl ..... 60  
 \MT@DisableLigatures ..... 2619  
 \MT@do@font ..... 557, 942, 951, 1255, 1418, 1506  
 \MT@documentfalse ..... 234  
 \MT@documenttrue ..... 234, 3723  
 \MT@dofalse ..... 823, 835, 856, 873, 900, 918  
 \MT@dotrue ..... 823, 826, 853, 870, 879, 897, 1771  
 \MT@draftfalse ..... 197, 3249  
 \MT@drafttrue ..... 197, 3251  
 \MT@edef@n ..... 283, 1948, 1974,  
     2304, 2738, 2788, 2790, 2921, 2936, 3068, 3275  
 \MT@encoding ..... 813, 910,  
     1907, 1919, 1999, 2083, 2114, 2131, 2138, 2146  
 \MT@encoding@check ..... 2934, 2938  
 \MT@end@catcodes ..... 1832, 1874, 2616,  
     2630, 2657, 2679, 2693, 2707, 2721, 2915, 3315  
 \MT@error ..... 49,  
     268, 603, 672, 1169, 2813, 2822, 3241, 3740  
 \MT@ex@c@name ..... 1226,  
     1228, 1309, 1310, 1320, 1326, 1327, 1339, 2663  
 \MT@ex@context ..... 783, 2271, 2309  
 \MT@ex@do ..... 1229, 1267  
 \MT@ex@doc@contexts ..... 2309  
 \MT@ex@factor ..... 202, 1243, 1313  
 \MT@ex@factor@ . 1243, 1257, 1260, 1279, 1280, 1307  
 \MT@ex@inh@name ..... 1291, 1292, 1294  
 \MT@ex@level ..... 202, 3582, 3584  
 \MT@ex@max ..... 214, 1282, 1283  
 \MT@ex@min ..... 214, 1285, 1286  
 \MT@ex@set@name ..... 2531  
 \MT@ex@split ..... 1269, 1273  
 \MT@exp@one@n .....  
     .. 291, 475, 599, 754, 756, 808, 848, 911,  
     1529, 1839, 2215, 2222, 2254, 2287, 2298,  
     2309, 2328, 2507, 2524, 2554, 2603, 2910, 3728

- \MT@exp@two@c ..... 293, 313, 320, 332, 510, 753, 787, 789,  
791, 802, 1607, 1626, 1629, 1998, 2003, 2084
- \MT@exp@two@n ..... 295, 866, 875, 2356, 2461
- \MT@expandfont ..... 1230, 1246, 1250
- \MT@expansion ..... 768, 1219, 3611
- \MT@expansionfalse ..... 193, 3484, 3576
- \MT@expansiontrue ..... 193, 3299
- \MT@extra@context ..... 2309,  
2649, 2663, 2686, 2699, 2713, 2760, 2763,  
2764, 2766, 2801, 2806, 2811, 2896, 3040,  
3044, 3047, 3050, 3053, 3057, 3058, 3060, 3087
- \MT@extra@inputenc ..... 2895, 2919, 2924, 2925
- \MT@factor@default ..... 226, 3280, 3494
- \MT@family ... 754, 813, 867, 910, 1908, 2595, 2596
- \MT@familyalias ..... 755,  
756, 819, 874, 876, 878, 880, 1918, 1920, 2597
- \MT@feat ..... 823, 962, 1058, 1060, 1062,  
1063, 1065, 1066, 1072, 1074, 1077, 1085–  
1089, 1091, 1093–1097, 1100, 1101, 1106,  
1110, 1113, 1116, 1119–1121, 1130, 1131,  
1136, 1154, 1609, 1805, 1807, 1810, 1811,  
1815, 1817, 1818, 1820, 1944, 1946, 1948,  
1952, 1956, 1960, 1961, 1964, 1972, 1974,  
1975, 1977, 1984, 1988, 2120, 2121, 2150, 2152
- \MT@features ..... 596,  
2309, 2322, 2507, 2518, 2548, 2785, 2905, 3324
- \MT@features@long ..... 596, 599, 604, 2284, 2917
- \MT@file@list ..... 1826,  
1828, 1836, 1839, 1841, 1846, 1849, 2603, 2607
- \MT@find@file ..... 754, 756, 1826
- \MT@fix@font@set ..... 2375, 3489
- \MT@font ..... 290,  
753, 778, 802, 809, 940, 941, 954, 958,  
965, 989, 990, 995, 996, 1016, 1017, 1038,  
1043, 1049, 1144, 1147, 1181, 1182, 1184,  
1185, 1251, 1255, 1289, 1290, 1334, 1336,  
1379, 1380, 1386, 1387, 1393, 1394, 1405–  
1407, 1409, 1410, 1415–1417, 1476, 1477,  
1483, 1484, 1495, 1496, 1498, 1499, 1504,  
1505, 1529, 1533, 1598, 1779, 2154, 2160,  
2191, 2192, 2194, 2215, 2217, 2223, 2244, 2255
- \MT@font@copy ..... 783, 784, 791, 792, 802–804
- \MT@font@list ..... 2154, 2215–2217, 2278
- \MT@font@orig ..... 784
- \MT@font@sets ..... 2360, 2375, 2465, 3489
- \MT@gdef@n ..... 281, 2642, 2656, 2678,  
2692, 2706, 2720, 2846, 2866, 2879, 2927, 3326
- \MT@get@axis ..... 2481–2484, 2494
- \MT@get@basefamily ..... 1838, 1875
- \MT@get@basefamily@ ..... 1881, 1884
- \MT@get@char@unit .... 980, 1108, 1134, 1369, 1466
- \MT@get@charwd ..... 1015, 1034, 1111, 1134
- \MT@get@config ..... 3303
- \MT@get@ex@opt ..... 1222, 1237, 1307
- \MT@get@ex@opt@ ..... 1315–1317, 1319, 1325
- \MT@get@font ..... 2460, 2470
- \MT@get@font@ ..... 2472, 2478, 2773
- \MT@get@font@and@size ..... 2759, 2771
- \MT@get@font@dimen ..... 1037, 1114
- \MT@get@font@dimen@six . 926, 957, 1347, 1444, 1543
- \MT@get@highlevel ... 2355, 2365, 2496, 2737, 2935
- \MT@get@inh@list ..... 929, 1224, 1350, 1447, 1970
- \MT@get@listname ..... 1890, 1946, 1972
- \MT@get@listname@ ..... 1890
- \MT@get@ls@basefont ..... 1578, 1619, 1622
- \MT@get@opt ..... 927, 1083, 1348, 1445
- \MT@get@range ..... 2384, 2394, 2747
- \MT@get@size ..... 2401, 2406, 2415, 2434, 2492
- \MT@get@slot .....  
978, 1276, 1367, 1464, 1786, 1991, 2959, 2965
- \MT@get@space@unit .....  
... 1108, 1209, 1377, 1384, 1391, 1474, 1481
- \MT@get@unit ..... 1116, 1124, 1554
- \MT@get@unit@ ..... 1124
- \MT@glet .. 279, 615, 778, 787, 1626, 1736, 2189,  
2279–2282, 2597, 2805, 2982, 3233, 3235, 3661
- \MT@gobble@settings ..... 957
- \MT@if@false ..... 191, 2227, 3085, 3759
- \MT@if@list@exists .....  
... 925, 1221, 1236, 1346, 1443, 1544, 1943
- \MT@if@true .....  
191, 2220, 3094, 3098, 3106, 3111, 3760–3764
- \MT@ifdefined@c@T .....  
299, 661, 662, 682, 686, 998, 1023, 1291,  
1396, 1486, 1593, 2595, 2924, 3029, 3336, 3339
- \MT@ifdefined@c@TF ..... 299, 629, 1570, 1581,  
1584, 1612, 1677, 1783, 1784, 1947, 1973, 3039
- \MT@ifdefined@n@T ..... 299,  
619, 845, 863, 891, 907, 999, 1119, 1155,  
1292, 1320, 1397, 1487, 1546, 1548, 1561,  
1563, 1565, 1926, 2149, 2342, 2590, 2666,  
2791, 2988, 2995, 3002, 3008, 3040, 3050, 3455
- \MT@ifdefined@n@TF ..... 299, 819,  
829, 846, 864, 892, 908, 1085, 1093, 1309,  
1326, 1773, 1810, 1914, 1999, 2533, 2537,  
2563, 3015, 3067, 3076, 3138, 3325, 3329, 3726
- \MT@ifdim .. 415, 530, 531, 535, 536, 2417, 2426,  
3092–3094, 3096, 3097, 3104–3106, 3109, 3110
- \MT@ifdimen .... 380, 2446, 2835, 2843, 2855, 3289
- \MT@ifempty ..... 335, 940, 941,  
987, 993, 1198, 1199, 1376, 1383, 1390,  
1415–1417, 1429–1431, 1433–1435, 1473,  
1480, 1504, 1505, 1521, 1522, 1658, 1659,  
1669, 1687, 1746, 2289, 2321, 2326, 2367,  
2373, 2395, 2396, 2409, 2410, 2471, 2486,  
2497, 2517, 2522, 2547, 2552, 2625, 2692,  
2772, 2782, 2787, 2801, 2804, 2811, 2864,  
2904, 2909, 2939, 3151, 3182, 3305, 3347, 3404
- \MT@ifint ..... 348, 2052, 2865, 3274, 3282
- \MT@ifstreq ..... 422,  
1161, 1806, 1945, 1960, 2100, 2103, 2253,  
2834, 2852, 2854, 3154, 3156, 3159, 3162,  
3184, 3186, 3200, 3201, 3257, 3258, 3263,  
3288, 3349, 3356, 3360, 3367, 3406, 3412, 3700
- \MT@in@clist ..... 496, 599, 620, 848, 866, 875,  
911, 1529, 1828, 1839, 2215, 2222, 2287, 2603
- \MT@in@rlist ..... 524, 894, 1935
- \MT@in@rlist@ ..... 524
- \MT@in@rlist@@ ..... 524
- \MT@in@tlist ..... 512, 1887, 2298
- \MT@in@tlist@ ..... 512
- \MT@inannotfalse ..... 72
- \MT@inannottrue ..... 72
- \MT@increment ..... 558, 2987, 2994, 3001, 3007

- `\MT@info` ..... 49, 64, 69, 2535,  
     3352, 3358, 3364, 3371, 3409, 3414, 3442, 3446  
`\MT@info@missing@char` ..... 1019, 1046  
`\MT@info@nl` ..... 49, 65, 66, 70,  
     630, 1048, 3255, 3312, 3330, 3333, 3487,  
     3493, 3500, 3583, 3612, 3627, 3632, 3637,  
     3641, 3647, 3651, 3679, 3685, 3691, 3736, 3747  
`\MT@inh@do` ..... 1983, 2950  
`\MT@inh@feat` ..... 2895, 2904, 2907, 2918  
`\MT@inh@split` ..... 2952, 2956  
`\MT@inlist@false` ..... 496, 500, 513, 525, 2230  
`\MT@inlist@true` ..... 496, 502, 520, 532, 537, 2230  
`\MT@is@active` ..... 1998, 2065  
`\MT@is@char` ..... 2003, 2084, 2090  
`\MT@is@composite` ..... 2001, 2111  
`\MT@is@feature` ..... 598, 2327, 2523, 2553  
`\MT@is@letter` ..... 1996, 2019, 2087, 2116  
`\MT@is@number` ..... 2035, 2040  
`\MT@is@symbol` ..... 2000, 2081  
`\MT@iterate` ..... 546  
`\MT@kerning` ..... 771, 1440, 3650  
`\MT@kerningfalse` ..... 199  
`\MT@kerningtrue` ..... 199  
`\MT@kn@cname` ..... 1449, 1451, 1512, 2713  
`\MT@kn@context` ..... 2272, 2309  
`\MT@kn@do` ..... 1452, 1455  
`\MT@kn@doc@contexts` ..... 2309  
`\MT@kn@factor` ..... 202  
`\MT@kn@factor@` ..... 1085  
`\MT@kn@inh@name` ..... 1486, 1487, 1489  
`\MT@kn@max` ..... 214  
`\MT@kn@min` ..... 214  
`\MT@kn@setname` ..... 2531  
`\MT@kn@split` ..... 1457, 1461  
`\MT@kn@split@val` ..... 1467, 1471  
`\MT@kn@unit` ..... 202  
`\MT@kn@unit@` ..... 1093, 1515  
`\MT@led@kern` ..... 626  
`\MT@led@unhbox@line` ..... 626  
`\MT@led@mac@setup` ..... 626, 678  
`\MT@let@cn` 288, 289, 820, 933, 1228, 1310, 1354,  
     1451, 1547, 1549, 1553, 1562, 1564, 1566,  
     1805, 1817, 1940, 1944, 1977, 2438, 2795, 2923  
`\MT@let@nc` ... 287, 625, 1121, 1655, 1956, 1984,  
     1988, 2225, 2278, 2344, 2346, 2352, 2457,  
     2671, 2832, 2836, 2841, 2844, 2850, 2853,  
     2856, 2885, 3069, 3074, 3160, 3163, 3363, 3370  
`\MT@let@nn` ..... 289,  
     1086, 1091, 1094, 1106, 1327, 1330, 2297, 3332  
`\MT@letterspace` .....  
     230, 1547, 1557, 1570, 1613, 3666, 3667, 3669  
`\MT@letterspace@` .....  
     . 950, 954, 1023, 1024, 1570–1572, 1575,  
     1579, 1583, 1590, 1612, 1613, 1683, 1687, 1745  
`\MT@letterspace@default` ..... 230, 3667  
`\MT@listname` .....  
     1890, 1937, 1940, 1947, 1948, 1950, 1952,  
     1973, 1974, 1976, 1977, 1981, 1984, 2968, 2974  
`\MT@load@inputenc` ..... 1162, 1165, 1175  
`\MT@load@list` ..... 931, 1226, 1352, 1449, 1803  
`\MT@loop` ..... 546, 553, 3073  
`\MT@lower` ..... 2388, 2394, 2750  
`\MT@ls@adjust` . 1599, 1615, 1742, 1743, 1751, 1766  
`\MT@ls@adjust@` ..... 1742, 1766  
`\MT@ls@basefont` ..... 1624–1626, 1629  
`\MT@ls@too@large` ..... 1749, 1754, 3669  
`\MT@ls@font` ..... 1574, 1579,  
     1590, 1591, 1594, 1598, 1607, 1614, 1636,  
     1638, 1655, 1660, 1664, 1671, 1681, 1685, 1688  
`\MT@lskern` ..... 1612, 1618, 1620, 1765, 1767  
`\MT@lslig` ..... 1617  
`\MT@lua` ..... 263, 353, 385, 427  
`\MT@map@clist@` ..... 468  
`\MT@map@clist@c` ... 468, 793, 1785, 2221, 2233,  
     2240, 2277, 2284, 2322, 2324, 2518, 2520,  
     2548, 2550, 2785, 2905, 2907, 2917, 3324, 3453  
`\MT@map@clist@n` ..... 468, 828,  
     1772, 2309, 2353, 2382, 2458, 2507, 2580,  
     2640, 2735, 2745, 2757, 2848, 2862, 2962,  
     3146, 3149, 3178, 3180, 3209, 3269, 3345, 3402  
`\MT@map@tlist@` ..... 486  
`\MT@map@tlist@c` .... 486, 515, 526, 796, 1000,  
     1293, 1398, 1488, 1894, 2246, 3041, 3086, 3489  
`\MT@map@tlist@n` ..... 486, 1867, 2092, 2981  
`\MT@maybe@do` .... 823, 923, 1219, 1344, 1441, 1531  
`\MT@maybe@etex` .... 235, 299, 324, 559, 1006, 1046  
`\MT@maybe@etex@` ..... 235  
`\MT@maybe@rem@from@list` ..... 2248, 2252  
`\MT@MT` ..... 46, 49, 52, 56, 58, 59,  
     64, 65, 67, 68, 152, 159, 270, 610, 673, 675,  
     2813, 3306, 3442, 3446, 3488, 3704, 3740, 3741  
`\MT@next@listname` ..... 1897, 1906  
`\MT@next@listname@` ..... 1915, 1927, 1933  
`\MT@nl@ligatures` ..... 1779, 2619  
`\MT@nl@setname` ..... 2619  
`\MT@noligatures` ..... 773, 1769, 3715  
`\MT@noligatures@` ..... 1636, 1638, 1779, 1782  
`\MT@noligaturesfalse` ..... 196  
`\MT@noligaturestrue` ..... 196, 2627  
`\MT@nonselectedfalse` ..... 1233, 1247  
`\MT@nonselectedtrue` ..... 1233, 1235  
`\MT@noresetfalse` 2018, 2025, 2034, 2055, 2100, 2103  
`\MT@noresettrue` ..... 1994, 2018, 2043  
`\MT@normal@catcodes` ..... 712, 1176, 1856, 1871  
`\MT@old@cmd` ..... 36, 41–44  
`\MT@opt@autofalse` ..... 3125  
`\MT@opt@autottrue` ..... 3125, 3211  
`\MT@opt@def@set` ..... 3137, 3165, 3190  
`\MT@opt@DVIfalse` ..... 3126  
`\MT@opt@DVIttrue` ..... 3126, 3214, 3217  
`\MT@opt@expansionfalse` ..... 3124  
`\MT@opt@expansiontrue` ..... 3124  
`\MT@options` ..... 2723  
`\MT@optwarn@admissible` ..... 3127, 3202, 3264  
`\MT@optwarn@nan` ..... 3131, 3276, 3284  
`\MT@orig@add@accent` ..... 2206  
`\MT@orig@foreign@language` ..... 3753, 3755  
`\MT@orig@pickupfont` .... 703, 733, 738, 2162, 2208  
`\MT@orig@py@macron` ..... 730, 734, 739  
`\MT@orig@select@language` ..... 3748, 3750  
`\MT@outer@space` 1600, 1649, 1694, 1701, 1713, 1716  
`\MT@pdf@annot` ..... 72  
`\MT@pdf@tex@no` ..... 96, 145, 150, 255  
`\MT@permute` 2655, 2677, 2691, 2705, 2719, 2930, 2978  
`\MT@permute@` ..... 2978  
`\MT@permute@@` ..... 2978

- \MT@permute@@@ . . . . . 2978
- \MT@permute@@@@ . . . . . 2978
- \MT@permute@@@@@ . . . . . 3006, 3011
- \MT@permute@@@@@@ . . . . . 3019, 3021
- \MT@permute@define . . . . . 3012, 3022–3024, 3065
- \MT@permute@reset . . . . . 2981, 3071
- \MT@permutelist . . . . .
  - . . . . . 2649, 2663, 2689, 2699, 2713, 2760,
  - 2761, 2766, 2928, 3035, 3036, 3040, 3044,
  - 3047, 3050–3053, 3060, 3061, 3087, 3118, 3119
- \MT@pickupfont . . . . . 735, 740, 2205, 2210
- \MT@pr@cname . . . . . 931, 933, 1189, 2649
- \MT@pr@context . . . . . 2271, 2309
- \MT@pr@do . . . . . 934, 969
- \MT@pr@doc@contexts . . . . . 2309
- \MT@pr@factor . . . . . 202, 3283, 3494, 3495
- \MT@pr@factor@ . . . . . 1085
- \MT@pr@inh@name . . . . . 998, 999, 1001
- \MT@pr@level . . . . . 202, 3492, 3493
- \MT@pr@max . . . . . 214
- \MT@pr@min . . . . . 214
- \MT@pr@setname . . . . . 2531
- \MT@pr@split . . . . . 971, 975
- \MT@pr@split@val . . . . . 981, 985
- \MT@pr@unit . . . . . 202, 3290, 3496
- \MT@pr@unit@ . . . . . 1093, 1192
- \MT@preset@aux . . . . . 1194,
  - 1196, 1198, 1199, 1202, 1517, 1519, 1521, 1522
- \MT@preset@aux@factor . . . . . 1194, 1202, 1429–1431, 1517
- \MT@preset@aux@space . . . . . 1196, 1202, 1433–1435, 1519
- \MT@preset@ex . . . . . 1321, 1338
- \MT@preset@kn . . . . . 1510
- \MT@preset@kn@ . . . . . 1510
- \MT@preset@pr . . . . . 1187
- \MT@preset@pr@ . . . . . 1187
- \MT@preset@sp . . . . . 1422
- \MT@preset@sp@ . . . . . 1422
- \MT@ProcessOptionsWithKV . . . . . 3449, 3469
- \MT@protrusion . . . . . 767, 923, 3499
- \MT@protrusionfalse . . . . . 192
- \MT@protrusiontrue . . . . . 192, 3296
- \MT@rbba@expansion . . . . . 591
- \MT@rbba@kerning . . . . . 591
- \MT@rbba@protrusion . . . . . 591
- \MT@rbba@spacing . . . . . 591
- \MT@rbba@tracking . . . . . 591
- \MT@register@font . . . . . 776, 2217, 2280
- \MT@register@font@cx . . . . . 2239, 2280
- \MT@register@subst@font . . . . . 2196, 2216, 2281
- \MT@register@subst@font@cx . . . . . 2232, 2281
- \MT@rem@from@clist . . . . . 507, 808, 2254, 2727
- \MT@rem@from@list . . . . . 798, 806
- \MT@repeat . . . . . 546, 555, 3079
- \MT@requires@luatex . . . . . 258, 350, 382, 424, 3621
- \MT@requires@pdftex . . . . . 254,
  - 348, 380, 422, 626, 750, 759, 770, 772,
  - 780, 946, 1007, 1021, 1258, 1343, 1440,
  - 1526, 1633, 1769, 2619, 2803, 2878, 2884,
  - 3230, 3298, 3399, 3539, 3562, 3615, 3695, 3712
- \MT@res@a . . . . . 445, 447
- \MT@res@b . . . . . 446, 447
- \MT@reset@context . . . . . 2262, 2265, 2269, 2296
- \MT@reset@context@ . . . . . 2269, 2296
- \MT@reset@ef@codes . . . . . 1223, 1245, 1258, 1322
- \MT@reset@ef@codes@ . . . . . 1253, 1261, 1265
- \MT@reset@kn@codes . . . . . 1446, 1453, 1501
- \MT@reset@kn@codes@ . . . . . 1501
- \MT@reset@pr@codes . . . . . 928, 935, 944
- \MT@reset@pr@codes@ . . . . . 944
- \MT@reset@sp@codes . . . . . 1349, 1356, 1412
- \MT@reset@sp@codes@ . . . . . 1412
- \MT@restore@catcodes . . . . . 178, 186
- \MT@restore@p@h . . . . . 659, 665, 666, 701, 710
- \MT@saved@setupfont . . . . . 3440, 3447
- \MT@scale . . . . . 566, 1010, 1059, 1073, 1076, 1280
- \MT@scale@factor . . . . . 1012, 1031, 1057, 1204, 1340
- \MT@scale@to@em . . . . . 988, 994, 1006, 1028,
  - 1210, 1378, 1385, 1392, 1475, 1482, 1556, 1678
- \MT@selectedfalse . . . . . 195
- \MT@selectedtrue . . . . . 195
- \MT@series . . . . . 813, 910, 1909, 1921
- \MT@set@all@ex . . . . . 1253, 1341
- \MT@set@all@kn . . . . . 1501, 1523
- \MT@set@all@pr . . . . . 937, 944, 1200
- \MT@set@all@sp . . . . . 1412, 1437
- \MT@set@babel@context . . . . . 3725, 3751, 3756, 3768
- \MT@set@default@set . . . . . 2548, 2554, 2561
- \MT@set@ex@codes . . . . . 1249, 3567, 3569
- \MT@set@ex@codes@n . . . . . 1233, 1249, 3569
- \MT@set@ex@codes@s . . . . . 1220, 3567
- \MT@set@ex@heirs . . . . . 1295, 1333
- \MT@set@inh@list . . . . . 2901, 2903
- \MT@set@inputenc . . . . . 930, 1152, 1225, 1351, 1448, 1982
- \MT@set@inputenc@ . . . . . 1155, 1157
- \MT@set@kn@codes . . . . . 1442
- \MT@set@kn@heirs . . . . . 1490, 1494
- \MT@set@listname . . . . .
  - 932, 1084, 1227, 1308, 1353, 1450, 1545, 2119
- \MT@set@lsfont . . . . . 1596, 1607
- \MT@set@named@keys . . . . . 2651, 2665, 2687, 2701, 2715, 2723
- \MT@set@pr@codes . . . . . 924, 947
- \MT@set@pr@heirs . . . . . 1002, 1180
- \MT@set@sp@codes . . . . . 1345
- \MT@set@sp@heirs . . . . . 1400, 1404
- \MT@set@tr@codes . . . . . 1540, 1610
- \MT@SetExpansion . . . . . 2661, 2663
- \MT@SetExtraKerning . . . . . 2711, 2713
- \MT@SetExtraSpacing . . . . . 2697, 2699
- \MT@SetProtrusion . . . . . 2647, 2649
- \MT@SetTracking . . . . . 2683, 2685
- \MT@setup@ . . . . . 612, 614, 615, 3224, 3225
- \MT@setup@contexts . . . . . 2261, 2276
- \MT@setupfont . . . . . 747, 2199, 3440, 3443, 3447, 3476
- \MT@setupfont@hook . . . . .
  - . . . . . 660, 679, 681, 685, 690, 701, 710, 717, 748
- \MT@shape . . . . . 813, 910, 1910, 1922
- \MT@shorthandoff . . . . . 3734, 3765, 3766
- \MT@show@pdfannot . . . . . 78
- \MT@shrink . . . . . 202, 1240, 3506,
  - 3508, 3510, 3514, 3515, 3518, 3522, 3572, 3585
- \MT@shrink@ . . . . . 1240, 1251, 1307
- \MT@shrink@default . . . . . 227, 3510
- \MT@size . . . . . 531, 535, 536, 813, 896, 899, 910
- \MT@size@name . . . . . 524, 1937
- \MT@sp@c@name . . . . . 1352, 1354, 1424, 2699
- \MT@sp@context . . . . . 2272, 2309, 3700



- \MT@sp@do ..... 1355, 1358  
 \MT@sp@doc@contexts ..... 2309  
 \MT@sp@factor ..... 202  
 \MT@sp@factor@ ..... 1085  
 \MT@sp@inh@name ..... 1396, 1397, 1399  
 \MT@sp@max ..... 214  
 \MT@sp@min ..... 214  
 \MT@sp@setname ..... 2531  
 \MT@sp@split ..... 1360, 1364  
 \MT@sp@split@val ..... 1370, 1374  
 \MT@sp@unit ..... 202  
 \MT@sp@unit@ ..... 1093, 1427  
 \MT@spacing ..... 771, 1343, 3640  
 \MT@spacingfalse ..... 198  
 \MT@spacingtrue ..... 198  
 \MT@split@name ..... 753, 813  
 \MT@step ..... 202,  
     1241, 3513, 3529, 3536, 3586, 3591, 3592, 3596  
 \MT@step@ ..... 1241, 1251, 1307  
 \MT@step@default ..... 227  
 \MT@stretch ..... 202, 1239, 3503, 3504,  
     3507, 3508, 3514, 3516, 3521, 3524, 3571, 3585  
 \MT@stretch@ ..... 1239, 1251, 1307  
 \MT@stretch@default ..... 227, 3504  
 \MT@temp ..... 939–942, 1414–1418,  
     1503–1506, 1651, 1655, 1660, 1663, 1668,  
     1701, 1704, 1707, 1710, 1713, 1716, 1721,  
     1724, 2480, 2505, 3452, 3456, 3461, 3463, 3466  
 \MT@tempencoding ..... 3014, 3015, 3025, 3032  
 \MT@tempfamily ..... 3026  
 \MT@tempseries ..... 3027  
 \MT@tempshape ..... 3028  
 \MT@tempsize ..... 2982, 3029, 3039, 3041, 3045  
 \MT@test@ast ..... 1676, 2366, 2371  
 \MT@textls ..... 20, 1742, 1743, 1745  
 \MT@tlist@break ..... 486, 521, 543, 1899, 3121  
 \MT@toks ..... 190, 1048, 1079, 1303,  
     1995, 2023, 2030, 2075, 2105, 2124, 2136, 2144  
 \MT@tr@adjust@space ..... 1587, 1650  
 \MT@tr@cc@name .....  
     ... 1546–1549, 1553, 1561–1566, 2688, 2692  
 \MT@tr@context ..... 2272, 2309  
 \MT@tr@doc@contexts ..... 2309  
 \MT@tr@factor@ ..... 1555  
 \MT@tr@font@list ..... 1527  
 \MT@tr@get@space ..... 1662, 1667, 1670, 1675  
 \MT@tr@ispace ..... 1563, 1581, 1582  
 \MT@tr@ligatures ..... 1561, 1593, 1635, 1638  
 \MT@tr@max ..... 214, 1755–1757  
 \MT@tr@min ..... 214, 1759–1761  
 \MT@tr@noligatures ..... 1593, 1632  
 \MT@tr@ospace ..... 1563, 1584, 1585  
 \MT@tr@outer@icr ..... 1707, 1726  
 \MT@tr@outer@icr@ ..... 1726  
 \MT@tr@outer@l ..... 1601, 1692  
 \MT@tr@outer@next ..... 1697, 1728  
 \MT@tr@outer@r ..... 1602, 1697, 1726  
 \MT@tr@outer@r@ ..... 1697  
 \MT@tr@setname ..... 2531  
 \MT@tr@unit@ ..... 1548, 1677  
 \MT@tracking .....  
     760, 1527, 1610, 2201, 3426, 3427, 3623, 3631  
 \MT@tracking@ ..... 1527, 3427  
 \MT@trackingfalse ..... 200  
 \MT@trackingtrue ..... 200  
 \MT@try@order ..... 1894, 1902  
 \MT@undefined@char ..... 2070, 2072, 2080  
 \MT@upper ..... 2388, 2394, 2750  
 \MT@use@set ..... 2518, 2524, 2531  
 \MT@val ..... 2289–2291, 2293, 2299, 2301,  
     2304, 2354, 2358, 2366, 2368, 2383–2385,  
     2397, 2400, 2402, 2405, 2407, 2408, 2414,  
     2416, 2417, 2419, 2422, 2424, 2435, 2438,  
     2442, 2446–2448, 2450, 2452, 2459, 2460,  
     2463, 2473–2475, 2488, 2490, 2495, 2497,  
     2499, 2501, 2736, 2738, 2746–2748, 2758,  
     2759, 2762, 2767, 2933, 2934, 2936, 2940,  
     2961, 2968, 2973, 2974, 3138, 3139, 3142,  
     3150, 3151, 3154, 3156, 3159, 3162, 3181,  
     3182, 3184, 3186, 3346, 3347, 3349, 3356,  
     3360, 3367, 3374, 3403, 3404, 3406, 3412, 3415  
 \MT@variants ..... 1887, 2573, 2583  
 \MT@vinfo ..... 49, 66, 766, 841, 1088, 1097,  
     1101, 1136, 1311, 1328, 1542, 1572, 1790,  
     1799, 1811, 1835, 1845, 1848, 1950, 1952,  
     1958, 2270, 2293, 2605, 2611, 3255, 3262, 3727  
 \MT@warn@ascii ..... 2032, 2123  
 \MT@warn@axis@empty ..... 2487, 2498, 2503  
 \MT@warn@code@too@large ..... 1063, 1066, 1070  
 \MT@warn@err ..... 49, 3259, 3260  
 \MT@warn@ex@too@large ..... 1283, 1286, 1301  
 \MT@warn@maybe@inputenc ..... 2137, 2145, 2148  
 \MT@warn@nodim ..... 2827, 2837, 2845, 2857  
 \MT@warn@number@too@large ..... 2059, 2129  
 \MT@warn@preset@twidth ... 1193, 1213, 1428, 1516  
 \MT@warn@rest ..... 2012, 2134  
 \MT@warn@tracking@DVI ..... 1573, 3654  
 \MT@warn@unknown ..... 2009, 2142  
 \MT@warning ..... 37, 49, 67, 1733, 1756,  
     1760, 1793, 1807, 1820, 1961, 2343, 2418,  
     2450, 2504, 2540, 2567, 2591, 2613, 2634,  
     2792, 2828, 2868, 2881, 2941, 3033, 3051,  
     3118, 3259, 3336, 3373, 3392, 3415, 3432–3434  
 \MT@warning@nl ..... 49, 61, 68, 149,  
     269, 609, 642, 651, 692, 959, 1039, 1077,  
     1129, 1214, 1302, 1643, 2124, 2130, 2135,  
     2143, 2175, 2668, 3128, 3133, 3142, 3260,  
     3291, 3319, 3473, 3531, 3542, 3553, 3573,  
     3594, 3622, 3656, 3676, 3682, 3688, 3701, 3770  
 \MT@while@num ..... 551, 557, 713–715  
 \MT@with@babel@and@T .....  
     ... 618, 661, 662, 680, 684, 3760–3764, 3766  
 \MT@with@package@T ..... 616, 663, 666,  
     671, 678, 688, 699, 709, 711, 724, 729, 2184  
 \MT@xadd ..... 76, 454, 1836, 1841, 1846,  
     1849, 2234, 2242, 2301, 2386, 2607, 2749, 2967  
 \MT@xaddb ..... 461, 2765, 3043  
 \MT@xdef@n ..... 283, 2534, 2538, 2565, 2570,  
     2594, 2760, 2783, 2893, 2925, 3060, 3139, 3141  
  
 N  
 \normalfont ..... 746, 3604, 3608  
 \nullfont ..... 746  
  
 P  
 \pdfadjustinterwordglue ..... 3428, 3636

<code>\pdfadjustspacing</code> .....	3398, 3582, 3607	<code>slantsc</code> (package) .....	147
<code>\pdfappendkern</code> .....	3430, 3646	<code>soul</code> (package) .....	49, 71, 177
<code>\pdfcopyfont</code> .....	791	<code>spacing</code> (option) .....	7, 11, 28, 106
<code>\pdfcprot</code> (package) .....	25, 47, 56, 83, 173	<code>\stbrcode</code> .....	1386, 1387, 1406, 1410, 1416
<code>\pdffontexpand</code> .....	1251	<code>step</code> (option) ...	8, 30, 108, 114, 116, 174, 176, 178
<code>\pdfnoligatures</code> .....	1792, 1798	<code>stretch</code> (option) ...	8, 15, 24, 30, 108, 114, 116, 178
<code>\pdfprependkern</code> .....	3429, 3645		
<code>\pdfprotrudechars</code> .....	3397, 3492	<b>T</b>	
<code>\pdftracingfonts</code> .....	257	<code>\tagcode</code> .....	1594, 1784, 1788
<code>\pickup@font</code> .....	703, 733, 735,	<code>tex4ht</code> (package) .....	47, 177
	738, 740, 2174, 2176, 2183, 2205, 2208, 2210	<code>\textls</code> .....	22, 23, 24, 705, 726, 1740
<code>pifont</code> (package) .....	83	<code>\textls*</code> .....	22
<code>pinyin</code> (package) .....	49, 178	<code>\textmicrotypecontext</code> .....	19, 21, 2268
<code>protrusion</code> (option) .....	6, 7, 11, 106	<code>tipa</code> (package) .....	83
<code>PSNFSS</code> (package) .....	169	<code>trace</code> (package) .....	84, 177
<code>pstricks</code> (package) .....	9, 114	<code>\tracingmicrotype</code> .....	60
<code>pxfonts</code> (package) .....	20, 121	<code>\tracingmicrotypeinpdf</code> .....	71
		<code>\tracingmicrotypeinpdfallfalse</code> .....	77
<b>Q</b>		<code>\tracingmicrotypeinpdfalltrue</code> .....	77
<code>qfonts</code> (package) .....	20, 121, 176	<code>tracking</code> (option) .....	7, 11, 22, 28, 106
		<code>txfonts</code> (package) .....	20, 121
<b>R</b>		<b>U</b>	
<code>relsize</code> (package) .....	90	<code>ucs</code> (package) .....	25, 80
<code>\rightmargin</code> .....	635	<code>ulgothic</code> (package) .....	20
<code>\rptcode</code> .....	941, 951, 995, 996, 1182, 1185	<code>unit</code> (option) .....	7, 29, 175–177
<b>S</b>		<code>\UseMicrotypeSet</code> .....	4, 11, 31, 43, 2337, 2515
<code>selected</code> (option) ....	8, 14, 15, 30, 61, 77, 107, 174	<b>V</b>	
<code>\selectfont</code> .....	1611, 2264, 2274, 3776	<code>verbose</code> (option) .....	9, 30, 108, 175
<code>\set@fontsize</code> .....	2442	<b>W</b>	
<code>\SetExpansion</code> .....	7, 14, 2659	<code>wordcount</code> (package) .....	46, 178
<code>\SetExtraKerning</code> .....	9, 18, 2709	<b>Y</b>	
<code>\SetExtraSpacing</code> .....	10, 17, 2695	<code>yfonts</code> (package) .....	23
<code>\SetProtrusion</code> .....	6, 12, 2645	<b>Z</b>	
<code>\SetTracking</code> .....	8, 16, 2681	<code>zefonts</code> (package) .....	20, 121
<code>\sfcode</code> .....	3699		
<code>\shbrcode</code> .....	1393, 1394, 1407, 1410, 1417		
<code>\showhyphens</code> .....	3602, 3606		
<code>shrink</code> (option) ..	8, 15, 30, 108, 114, 116, 176, 178		

## 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

### Preamble

The L<sup>A</sup>T<sub>E</sub>X Project Public License (LPPL) is the primary license under which the the L<sup>A</sup>T<sub>E</sub>X kernel and the base L<sup>A</sup>T<sub>E</sub>X packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is T<sub>E</sub>X-related (such as a L<sup>A</sup>T<sub>E</sub>X package), but it is written in such a way that you can use it even if your work is unrelated to T<sub>E</sub>X.

The section ‘WHETHER AND HOW TO DISTRIBUTE

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

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 per-

son 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.

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.

### 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

### 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.