



# 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)

v1.9a — 2005/12/05

## Abstract

The microtype package provides an interface to the micro-typographic extensions of pdf $\TeX$ : most prominently, character protrusion and font expansion, furthermore the possibility to disable all ligatures of a font.<sup>1</sup> It allows to apply these features to customizable 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>2</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 require pdf $\TeX$  1.30. The package will by default enable the features that can safely be assumed to work.

- 
- 1 A preview of the next version with support for even more micro-typographical extensions is also included in this package. Footnote 11 on page 17 contains the details.
  - 2 Currently, this package provides settings for Computer Modern Roman, Palatino, Times, Adobe Garamond and Minion, Bitstream Charter, and the AMS math 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>Invoking the Package</b>	<b>5</b>
<b>3</b>	<b>Options</b>	<b>5</b>
3.1	Micro-Typographic Options . . . . .	5
3.2	Options for Character Protrusion . . . . .	6
3.3	Options for Font Expansion . . . . .	6
3.4	Miscellaneous Options . . . . .	7
3.5	Changing Options Later . . . . .	8
<b>4</b>	<b>Declaring Font Sets</b>	<b>8</b>
<b>5</b>	<b>Micro Fine Tuning</b>	<b>11</b>
5.1	Character Protrusion . . . . .	11
5.2	Font Expansion . . . . .	13
5.3	Character Inheritance . . . . .	15
5.4	Configuration Files . . . . .	15
<b>6</b>	<b>Disabling Ligatures</b>	<b>16</b>
<b>7</b>	<b>Context-sensitive Setup</b>	<b>17</b>
<b>8</b>	<b>Hints and Caveats</b>	<b>17</b>
<b>9</b>	<b>Contributions</b>	<b>19</b>
<b>10</b>	<b>Acknowledgments</b>	<b>20</b>
<b>11</b>	<b>References</b>	<b>20</b>
<b>12</b>	<b>Short History</b>	<b>21</b>
<b>13</b>	<b>Implementation</b>	<b>24</b>
13.1	Requirements . . . . .	26
13.2	Compatibility . . . . .	29
13.3	Auxiliary macros . . . . .	32
13.4	Setting up a font . . . . .	39
	[Protrusion – 42] [Expansion – 48] [Interword Space (Glue) – 51]	
	[Additional Kerning – 53] [Letter spacing – 55] [Disabling	
	Ligatures – 56] [Loading the Configuration – 56] [Translating Characters	
	into Slots – 61] [Hook into $\LaTeX$ 's font selection – 66] [Context-sensitive	
	Setup – 68]	
13.5	Configuration . . . . .	71
	[Font Sets – 71] [Interaction with babel – 78] [Fine Tuning – 79]	
	[Character Inheritance – 85] [Permutation – 88]	

13.6 Changing Options Later . . . . .	91
13.7 Package Options . . . . .	94
[Declaring the Options – 94] [Hook for Other Packages – 98] [Processing the Options – 99]	
<b>14 Configuration Files</b>	<b>106</b>
14.1 Font Sets . . . . .	106
14.2 Interaction with babel . . . . .	107
14.3 Note on Admissible Characters . . . . .	107
14.4 Character Inheritance . . . . .	108
[OT1 – 108] [T1 – 108] [LY1 – 109] [OT4 – 110] [T5 – 110]	
14.5 Font Expansion . . . . .	111
14.6 Character Protrusion . . . . .	112
[Default – 113] [Italics – 119] [Small Caps – 126] [Italic Small Caps – 129]	
[textcomp – 130] [Math – 134] [AMS fonts – 140]	
14.7 Interword Spacing . . . . .	145
14.8 Additional Kerning . . . . .	148
<b>15 Auxiliary File for Micro Fine Tuning</b>	<b>150</b>
<b>A Change History</b>	<b>152</b>
<b>B Index</b>	<b>158</b>

## List of Tables

1 Predefined font sets . . . . .	10
2 Fonts with tailored protrusion settings . . . . .	16
3 Order for matching font attributes . . . . .	59

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

pdf<sub>T</sub><sub>E</sub>X, the <sub>T</sub><sub>E</sub>X extension written by Hàn Thế Thành, introduces two 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: *character protrusion* (also known as margin kerning) and *font expansion*. Quoting Hàn Thế Thành's thesis:

'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 L<sup>A</sup><sub>T</sub><sub>E</sub>X user interface for quite some time. Then, the pdfcprot package was released [pdfcprot], which allowed L<sup>A</sup><sub>T</sub><sub>E</sub>X users to employ character protrusion without having to mess much with the internals.

Font expansion, however, was still most difficult to utilize, since it required that the font metrics are available in all levels of expansion. Therefore, anybody who wanted to use 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 can be expected to prove as a major facilitation for <sub>T</sub><sub>E</sub>X and L<sup>A</sup><sub>T</sub><sub>E</sub>X users: Font expansion can now take place automatically. That is, pdf<sub>T</sub><sub>E</sub>X no longer needs the expanded font metrics but will calculate them at run-time, and completely in memory.

Finally, the possibility to disable all ligatures of a font has been introduced. This may be useful when using typewriter fonts.

The microtype package provides an interface to all these micro-typographic extensions.<sup>3</sup> All micro-typographic aspects may be customized to your taste and needs in a straightforward manner. The next chapters will present a survey of all options and customization possibilities.

<sup>3</sup> Therefore, it is an alternative, not a supplement, to the pdfcprot package, which provides an interface to character protrusion.

## 2 Invoking the Package

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.

## 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 3.4).

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

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

```
\usepackage[protrusion,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.

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.

Whether ligatures should be disabled cannot be controlled via a package option but by using the `\DisableLigatures` command, which is explained in section 6.

### 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](#), p. 20). If expanded instances of the fonts are available, they will be used regardless whether auto is true or not.

Automatic font expansion requires fonts in Type 1 format. Therefore, if you are using the Computer Modern Roman fonts in T1 encoding<sup>4</sup>, you should either install the `cm-super` fonts or use the Latin Modern fonts (package `lmodern`).

**stretch** *<integer>* 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.

<sup>4</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'.

**step** *<integer>* min(stretch,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 Miscellaneous Options

**DVIoutput** true, false false

pdfTeX is not only able to generate PDF output but can also spit out DVI files.<sup>5</sup> The latter can be ordered with the option `DVIoutput`, which will set `\pdfoutput` to zero.

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

When generating DVI files, font expansion has to be enabled explicitly. *Automatic* font expansion will not work because `dvips` (resp. the DVI viewer) is not able to generate the expanded 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*. The `draft` and `final` options may also be inherited from the class options; of course, you can override them in the package options.

<sup>5</sup> TeX systems are beginning to switch to pdfTeX as the default engine even for DVI output.

**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.

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

Various settings for this package will be loaded from a main configuration file, by default `microtype.cfg` (see section 5.4). You can have a different configuration file loaded instead by specifying its name *without the extension*, e.g., `config=mycrottype`.

### 3.5 Changing Options Later

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

Inside the preamble, this command accepts all package options described above (except for `config`).

In the document body, this command may be used to change the general settings of the micro-typographic extensions. It then accepts the keys: **expansion**, **protrusion** and **activate**, which in turn may receive the values true, false, compatibility or nocompatibility (but not the name of a font set). Using this command, you could for instance temporarily disable font expansion by saying:

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

## 4 Declaring Font Sets

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 expanded. You may want to customize the set of fonts that should get the benefit of micro-typographic treatment. This can be achieved by specifying attributes of the font that have to be matched for them to be taken into account.

`\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 set can subsequently be activated by calling:

```
\UseMicrotypeSet[<features>]{<set name>}
```

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 'basic`text`' in the main configuration file as follows:

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

If you now call

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

in the document's preamble, only fonts in the text encodings T1, OT1, LY1, OT4 or T5, roman or sans serif families, normal (or 'medium') series, and in sizes called by `\normalsize`, `\footnotesize`, `\small` or `\large`, will be expanded. 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 'all`text`', which is declared as:

```
\DeclareMicrotypeSet{alltext}
{ encoding = {T1,OT1,LY1,OT4,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 example above), it does not designate an NFSS code, but will expand to the document's `\<value>default`, e.g. `\rmdefault`. For example, if you want to include the bold font, too, you should say 'bf\*' instead of 'b' (`\bfdefault` for Computer Modern is 'bx', while for other fonts, it might be 'b' or even 'sb'). A single asterisk means `\<characteristic>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'); 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 characteristics, separated by forward slashes, i.e., 'font = `<encoding>/<family>/<series>/<shape>/<size>`'. This allows you to add fonts to the set that are otherwise disjunct from it. For instance, if you wanted to have the roman family in all sizes protruded, but only the normal

Table 1: Predefined font sets

Set name	Font attributes				
	Encoding	Family	Series	Shape	Size
<code>all</code>	–	–	–	–	–
<code>alltext</code> ( <code>allmath</code> )	OT1, T1, LY1, OT4, T5, TS1 (OML, OMS, U)	–	–	–	–
<code>basictext</code> ( <code>basicmath</code> )	OT1, T1, LY1, OT4, T5 (OML, OMS)	<code>\rm*</code> , <code>\sf*</code>	<code>m</code>	–	<code>\normalsize</code> , <code>\footnotesize</code> , <code>\small</code> , <code>\large</code>
<code>normalfont</code>	<code>\encoding*</code>	<code>\family*</code>	<code>\series*</code>	<code>\shape*</code>	<code>\normalsize</code>

‘\*’ = ‘default’

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 six predefined font sets. They may also be activated by passing their name to the feature options `expansion` and `protrusion` when loading the package, for example:

```
\usepackage[protrusion=allmath,expansion=basimath]{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 the package has been loaded without activating any font sets, the sets declared by this command will be activated. By default, the ‘`basictext`’ font set will be used for font expansion, the ‘`alltext`’ set for character protrusion.

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 amount of protrusion. 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 customizing these finer aspects of micro-typography.

### 5.1 Character Protrusion

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

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

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

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

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

The *characters* may be specified either as a single character (‘A’), as a text symbol command (‘\textquoteleft’), or as a slot number: three digits for decimal notation, prefixed with ‘#’ for hexadecimal, with ‘o’ for octal (e. g., the ‘fl’ ligature in T1 encoding: 029, #1D, o35). 8-bit characters may be entered directly or in the L<sup>A</sup>T<sub>E</sub>X 7-bit way of defining them: both ‘Ä’ and ‘\A’ are valid, provided the character is actually included in the encoding(s). You also have the possibility to declare lists of characters that should inherit protrusion or expansion factors (see section 5.3).

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.

The set of fonts to which the settings should apply is declared using the same syntax of  $\langle \text{font axis} \rangle = \langle \text{value list} \rangle$  pairs as for the command `\DeclareMicrotypeSet`. 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>6</sup> The encoding must always match.

*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 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>7</sup>

<sup>6</sup> For the interested, table 3 on page 59 presents the exact order.

<sup>7</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.

**preset** Presets the protrusion codes of all characters to the specified values ( $=\{\langle left \rangle, \langle right \rangle\}$ ), possibly scaled by a factor. A unit setting will only be taken into account if it is not `=character`.

**context** The scope of the list may be limited to a certain context. For an example application, see section 7.

## 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  $\langle character \rangle = \langle expansion factor \rangle$  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.

The *set of fonts* is declared in the same way as for `\SetProtrusion`.

*Options:*

**name, load, preset, 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, or to determine the context of the list.

**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. You could take advantage of the `stretch` and `shrink` options to allow for more expansion in this particular paragraph. There is one problem that has to be worked around, however: pdf<sub>T</sub><sub>E</sub>X prohibits the use of the same font with different expansion parameters. 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 could exploit a dirty trick and load a minimally larger font for the paragraph in question. E. g., for a document printed in 10pt:<sup>8</sup>

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

**factor** This option provides a different method to alter expansion settings for certain fonts, working around another restriction of pdfTeX: It does not allow different expansion limits or steps within one paragraph. 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.

<sup>8</sup> 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.

### 5.3 Character Inheritance

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

In most cases, accented characters should inherit the protrusion resp. expansion factors from the respective base character. For example, all of the characters À, Á, Â, Ã, Ä, Å and Æ should probably be protruded by the same (absolute) amount as the character A. Using the command `\DeclareCharacterInheritance`, you may declare such lists of characters, so that you then only have to set up the base characters. With the optional argument, which may contain a comma-separated list of features, you may confine the scope of the list. The font set can be declared in the usual way, with the only exception that you must specify exactly one encoding. The inheritance lists are to be 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.

### 5.4 Configuration Files

The default configuration, consisting of inheritance settings, declarations of font sets and alias fonts, and generic protrusion and expansion 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 3.4).

If you are embarking on creating new expansion and protrusion settings for a font family, you should put them into a separate file, whose name must be: ‘`mt-font family.cfg`’ (e.g. ‘`mt-pad.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. If the font name consists of four characters, the package will also try to find the file for the base font family by removing the suffix denoting the sub-family, so that you may put settings for the fonts `padx` (expert set), `padj` (oldstyle numerals) and `pad` (plain) into one and the same file.

This package ships with configuration files for the font families Computer Modern Roman, Palatino, the inescapable Times, Adobe Garamond and Minion<sup>9</sup>, for Bitstream Charter and the AMS math fonts. Table 2 lists them all.

If you have created a file for another font and you are willing to share, don’t hesitate to send it to me so that it can be included in future releases of this package.

`\DeclareMicrotypeAlias` {*font name*} {*alias font*}

You may use this command for fonts that are very similar, or actually the same (for instance if you did not stick to the Berry naming scheme when installing the font). An example would be the Latin Modern fonts which are clones of the Computer

<sup>9</sup> By courtesy of Harald Harders ([h.harders@tu-bs.de](mailto:h.harders@tu-bs.de)).

Table 2: Fonts with tailored protrusion settings

Font family (NFSS code)	Features	
	Encodings	Shapes
Generic	OT1, T1, LY1, (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
Adobe Minion (pmnx, pmnj)	OT1, T1, LY1, TS1	n, it, (sl) <sup>d</sup> , sc, si
Palatino (ppl, pplx, pplj) <sup>e</sup>	OT1, OT4, T1, LY1, (TS1) <sup>a</sup>	n, it, (sl) <sup>d</sup> , sc
Times (ptm, ptmx, ptmj) <sup>f</sup>	OT1, OT4, T1, LY1, (TS1) <sup>a</sup>	n, it, (sl) <sup>d</sup> , sc
Computer Modern math fonts (cmsy, cmm)	OML/OMS	n/it
AMS math fonts (msa, msb, euf, eus)	U	n

<sup>a</sup> Incomplete

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

<sup>c</sup> Also used for: mathdesign/Charter (mdbch)

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

<sup>e</sup> Also used for: pxfonts (pxr), qfonts/QuasiPalatino (qpl)

<sup>f</sup> Also used for: txfonts (txr), qfonts/QuasiTimes (qtm)

Modern fonts, so that it is not necessary to create new settings for them – you could say:

```
\DeclareMicrotypeAlias{lmr}{cmr}
```

which would make the package, whenever it encounters the font lmr and does not find settings for it, also try the font cmr. In fact, you will find this very line in the default configuration file, along with others for the virtual fonts provided by the packages ae, zefonts, eco and hfoldsty.

```
\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>10</sup> This command will load the file mt-*<font name>*.cfg.

## 6 Disabling Ligatures

```
\DisableLigatures {<set of fonts>}
```

A new feature has been introduced with pdfTeX 1.30: The possibility to completely disable all ligatures of a font (which will also switch off kerning). While this purposely *lowers* the micro-typographic quality instead of raising it, it is especially

<sup>10</sup> Font package authors might also want to have a look at the hook \Microtype@Hook, described in the implementation part, section 13.7.2.



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* }
```

## 7 Context-sensitive Setup

In previous versions of microtype, each font was set up exactly once for the entire document. Since version 1.9, it is possible to apply different settings to a font depending on the context it appears in.

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

This command may be used anywhere in the document (also in the preamble) to change the micro-typographic context. For each feature (*protrusion*, *expansion*), one context may be specified. Only settings which have been specified with the corresponding ‘context’ keyword will then be applied. This makes it possible to use different settings for different parts of the document.<sup>11</sup>

## 8 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

<sup>11</sup> This feature is especially useful for the new experimental extensions of pdfTeX: adjustment of interword spacing (glue) and the possibility to specify additional character kerning. The former may improve the appearance of the text even more, the latter allows for instance to insert small spaces before certain characters (e. g., for typesetting in the French tradition) without having to use active characters; also, letterspacing can be implemented in a robust way. Currently, these extensions are only available through patches from <http://pdfTeX.sarovar.org/>. However, if you are adventurous, know how to apply the patches and you are able to compile pdfTeX yourself, you can easily experiment with them, since microtype already supports these new extensions. To generate the extended version of the microtype package and its documentation, simply remove the comments before ‘\betatrue’ near the beginning of `microtype.ins` and `microtype.dtx`.

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.* Because each expanded instance of the font will be embedded in the PDF file, 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.

*Compatibility.* The package should 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 aware of the following issues:

- When using 8-bit characters in the configuration, `inputenc` must be loaded first. Unicode input in the configuration is currently not supported.
- The CJK package, like `microtype`, hooks into the  $\text{\LaTeX}$  font selection scheme. Therefore, both packages probably don't cooperate well. However, since I know nothing about CJK, I would appreciate feedback on the interaction of both packages – be it positive or negative.

*You might want to disable protrusion in verbatim environments.* As you know by now, `microtype` will by default apply character protrusion to all fonts part of 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 incantation, added to the document's preamble, would serve the same purpose:<sup>12</sup>

```
\makeatletter
\g@addto@macro\verbatim{\pdfprotrudechars=0 \pdfadjustspacing=0\relax}
\makeatother
```

<sup>12</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.

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

- Warning: pdflatex: font ptmr8r cannot be expanded (not an included Type1 font)  
Font expansion can only be applied if the font is actually embedded in the PDF file. If you receive the above error message, your T<sub>E</sub>X system is not set up to embed (or ‘download’) the base PostScript fonts (e. g. Times, Helvetica, Courier). In most T<sub>E</sub>X distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to true. Otherwise, consult the local guide of your T<sub>E</sub>X system.
- Warning: pdflatex (file ecrm1000+20): Font ecrm1000+20 at 1200 not found  
Furthermore, automatic font expansion requires Type 1 fonts. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap font. This is not possible, unless you manually create expanded instances of the fonts.
- ! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.  
Such an error message could occur if you are trying to employ font expansion while creating DVI output. Remember, that *automatic* font expansion only works when running pdfT<sub>E</sub>X in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your T<sub>E</sub>X system.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf\_mem\_size)=65536].  
When applying micro-typographic enhancement to a large document with a lot of fonts, you may be running out of pdfT<sub>E</sub>X memory. The memory can be increased by setting `pdf_mem_size` to a larger value (maximum 524 288). For teT<sub>E</sub>X-based systems, change the settings in `texmf.cnf`, for MiK<sub>T</sub>E<sub>X</sub>, in the file `miktex.ini`. Beginning with version 1.30 of pdfT<sub>E</sub>X, memory will grow dynamically, so that this problem can no longer arise.

## 9 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).<sup>13</sup>

<sup>13</sup> Should you have lots of pdfcprot configuration files lying around, I can also provide you with a T<sub>E</sub>X conversion script. Just ask me.

## 10 Acknowledgments

This package would be pointless if *Hàn Thế Thành* hadn't created the pdf $\TeX$  programme in the first place, which introduced the micro-typographic extensions and made them available to the  $\TeX$  world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004].

*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.

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 or helped otherwise (in chronological order): *Ulrich Dirr*, *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* and *Ralf Stubner*.

## 11 References

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

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

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, *The pdf $\TeX$  user manual*, August 21, 2005. (Available from CTAN at [/systems/pdftex/manual](#); latest version at <http://sarovar.org/projects/pdftex/>)

$\LaTeX$ 3 Project Team,  *$\LaTeX$ 2 $\epsilon$  font selection*, February 10, 2004. (Available from CTAN at [/macros/latex/doc/fntguide.pdf](#))

Carsten Schurig, *The pdfcprot.sty package*, August 14, 2002. (Available from CTAN at [/macros/latex/contrib/pdfcprot/](#))

## 12 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 fixes are swept under the rug.

### 1.9a (5. 12. 2005)

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  
 Protrusion settings for T5 encoded Charter

### 1.9 (28. 10. 2005)

New command `\DisableLigatures` to disable ligatures of fonts (requires pdfTeX version 1.30 or later; see section 6)  
 New command `\microtypecontext` to change the configuration context; new key ‘context’ for the configuration commands (see section 7)  
 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)  
 ‘DVIOutput’ option will work with TeXLive 2004

### 1.8 (23. 6. 2005)

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 3.4)  
 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 files 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 (23. 3. 2005)

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 5.4)
- Hook `\Microtype@Hook` for font package authors (see section 13.7.2)
- New option ‘`verbose=errors`’ to turn all warnings into errors
- Disable expansion inside `\showhyphens`
- Warning when running in draft mode
- 1.6a (2. 2. 2005)
  - Compatibility with the `frenchpro` package
- 1.6 (24. 1. 2005)
  - 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 (15. 12. 2004)
  - 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`’:  $4 \cdot (\min(\text{stretch}, \text{shrink})/5)$  (see section 3.3)
  - Protrusion settings for Bitstream Charter
  - Compatibility with Turkish `babel`
- 1.4b (26. 11. 2004)
  - `\UseMicrotypeSet` requires the set to be declared (see section 4)
- 1.4 (12. 11. 2004)
  - 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 3.5)
  - New option: ‘`final`’
- 1.3 (27. 10. 2004)
  - Compatibility with the `german` and `ngerman` packages
- 1.2 (3. 10. 2004)
  - 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 5.4)
  - More tests for sanity of settings and whether all fonts will be set up
  - More robust parsing of sizes in font sets

### 1.1 (21. 9. 2004)

Protrusion settings for Adobe Minion, contributed by Harald Harders

New command: `\DeclareCharacterInheritance` (see section [5.3](#))

Characters may also be specified as octal or hexadecimal numbers (see section [5](#))

Configuration file names in lowercase (see section [5.4](#))

### 1.0 (11. 9. 2004)

First CTAN release

## 13 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.

config: Surrounds all configuration modules.

m-t: The main configuration file (microtype.cfg).

bch: Settings for Bitstream Charter (mt-bch.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.

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 AMS Euler Fraktur font (mt-euf.cfg).

eus: Settings for AMS Euler script font (mt-eus.cfg).

test: A helper file that may be used to create and test protrusion settings (test-microtype.tex).

beta: Support for features not yet included in an official release of pdfTeX.

And now for something completely different.

1 *<package>*

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

```

2 \newcommand*\DeclareMicrotypeSet[3] [] {}
3 \newcommand*\UseMicrotypeSet[2] [] {}
4 \newcommand*\DeclareMicrotypeSetDefault[2] [] {}
5 \newcommand*\DeclareMicrotypeAlias[2] {}
6 \newcommand*\SetProtrusion[3] [] {}
7 \newcommand*\SetExpansion[3] [] {}
8 \newcommand*\DisableLigatures[1] {}
9 \newcommand*\DeclareCharacterInheritance[3] [] {}
10 \newcommand*\LoadMicrotypeFile[1] {}
11 \newcommand*\microtypesetup[1] {}
12 \newcommand*\microtypecontext[1] {}
13 <beta>
14 \newcommand*\SetExtraSpacing[3] [] {}

```



```

15 \newcommand*\SetExtraKerning[3] [] {}
16 \newcommand*\DeclareMicrotypeBabelHook[2] {}
17 \newcommand\textls[2] [] {#2}
18 \newcommand\lsstyle{}
19 /beta

```

This command also has a starred version.

```
20 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
```

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.

```

21 \@onlypreamble{\DeclareMicrotypeSet}
22 \@onlypreamble{\UseMicrotypeSet}
23 \@onlypreamble{\DeclareMicrotypeSetDefault}
24 \@onlypreamble{\DisableLigatures}
25 /beta \@onlypreamble{\DeclareMicrotypeBabelHook}

```

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

```

26 \def\MT@old@cmd#1#2{\MT@warning{%
27   \string#1 is deprecated. Please use\MessageBreak
28   \string#2 instead}%
29   \let#1#2#2}

30 \newcommand*\DeclareMicroTypeSet{%
31   \MT@old@cmd\DeclareMicroTypeSet
32   \DeclareMicrotypeSet}
33 \newcommand*\UseMicroTypeSet{%
34   \MT@old@cmd\UseMicroTypeSet
35   \UseMicrotypeSet}
36 \newcommand*\DeclareMicroTypeAlias{%
37   \MT@old@cmd\DeclareMicroTypeAlias
38   \DeclareMicrotypeAlias}
39 \newcommand*\LoadMicroTypeFile{%
40   \MT@old@cmd\LoadMicroTypeFile
41   \LoadMicrotypeFile}

```

`\MT@MT` This is us.

```
42 \def\MT@MT{microtype}
```

`\MT@error` Communicate.

```

\MT@warning 43 \def\MT@error{\PackageError{\MT@MT}}
\MT@warning@n1 44 \def\MT@warning{\PackageWarning{\MT@MT}}
\MT@warn@err 45 \def\MT@warning@n1#1{\MT@warning{#1\@gobble}}
\MT@info 46 \def\MT@warn@err#1{\MT@error{#1}{%
\MT@info 47   This error message appears because you loaded the '\MT@MT'\MessageBreak
\MT@info@n1 48   package with the option 'verbose=errors'. Consult the documentation\MessageBreak
\MT@vinfo 49   in \MT@MT.(pdf,dvi) to find out what went wrong.}}
50 \def\MT@info{\PackageInfo{\MT@MT}}
51 \def\MT@info@n1#1{\MT@info{#1\@gobble}}
52 /!debug \let\MT@vinfo\@gobble

```

Debug. Cases for `\tracingmicrotype`:

0: almost none

1: + sets & lists

2: + heirs

3: + slots

4: + factors

```

53 <*debug>
54 \let\MT@vinfo\MT@info@n1
55 \newcount\tracingmicrotype
56 \tracingmicrotype=\tw@
57 \def\MT@info#1#2{\ifnum\tracingmicrotype<#1\relax\else\MT@info{#2}\fi}
58 \def\MT@info@n1#1#2{\ifnum\tracingmicrotype<#1\relax\else\MT@info@n1{#2}\fi}
59 </debug>

```

## 13.1 Requirements

\MT@pdftex@no pdfTeX'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 pdfTeX we're using, if any. \MT@pdftex@no will be used throughout the package to respectively do the right thing.

Currently, there are six cases for pdfTeX:

0: not running pdfTeX

1: pdfTeX (< 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 ( $\geq$  1.30)

6: + adjustment of interword spacing; extra kerning; \pdfmatch<sup>14</sup> ( $\geq$  1.4x)

```

60 \let\MT@pdftex@no\z@

```

A hack circumventing the TeXLive 2004 hack which undefines the pdfTeX primitives in the format in order to hide the fact that pdfTeX is being run from the user. This has been *fixed* in TeXLive 2005.

```

61 \ifx\normalpdftexversion\undefined \else
62   \let\pdftexversion\normalpdftexversion
63   \let\pdftexrevision\normalpdftexrevision
64   \let\pdfoutput\normalpdfoutput
65 \fi

```

Old packages might have defined \pdftexversion to \relax.

```

66 \ifx\pdftexversion\undefined \else
67   \ifx\pdftexversion\relax \else
68     <debug>\MT@info@n1{0}{running pdftex \the\pdftexversion(\pdftexrevision)}
69     <beta> \ifx\pdfadjustinterwordglue\undefined
70       \def\MT@pdftex@no{5}
71     <beta> \else \def\MT@pdftex@no{6}\fi
72     \ifnum\pdftexversion < 130
73       \def\MT@pdftex@no{4}
74     \ifnum\pdftexversion < 120
75       \let\MT@pdftex@no\thr@@

```

<sup>14</sup> This command was actually introduced in 1.30, however, pdfTeX ran in a buffer overflow with strings larger than 1024 bytes.

```

76     \ifnum\pdfTeXversion = 14
77     \ifnum \expandafter'\pdfTeXrevision < 'h
78         \let\MT@pdfTeX@no\tw@
79         \ifnum \expandafter'\pdfTeXrevision < 'f
80             \let\MT@pdfTeX@no@ne
81         \fi
82     \fi
83 \else
84     \ifnum\pdfTeXversion < 14
85         \let\MT@pdfTeX@no@ne
86     \fi
87 \fi
88 \fi
89 \fi
90 \fi
91 \fi
92 (debug)\MT@info@nl{0}{pdfTeX no: \number\MT@pdfTeX@no}

```

\MT@requires@pdfTeX For definitions that depend on a particular pdfTeX version.

```

93 \def\MT@requires@pdfTeX#1{%
94     \ifnum\MT@pdfTeX@no<#1\relax
95         \expandafter\@secondoftwo
96     \else
97         \expandafter\@firstoftwo
98     \fi
99 }

```

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

```

100 \ifnum\MT@pdfTeX@no<\tw@
101 \AtEndOfPackage{\let\@unprocessedoptions\relax}%
102 \let\CurrentOption\@empty
103 \MT@warning@nl{%
104     \ifcase\MT@pdfTeX@no
105         You don't seem to be using pdfTeX.\MessageBreak
106     \or
107         You are using a pdfTeX version older than 0.14f.\MessageBreak
108         '\MT@MT' won't work with such antiquated versions.\MessageBreak
109         Please install a newer version of pdfTeX.\MessageBreak
110     \fi
111     All micro-typographic features will be disabled}
112 \expandafter
113 \endinput
114 \fi

```

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.

```

115 \def\MT@catcodes{%
116     \catcode'\^7 %
117     \@makeother\~%
118     \@makeother\=%
119     \@makeother\*%
120     \@makeother\,%
121     \@makeother\/%
122     \@makeother\'%

```

```

123 \@makeother\'%
124 \@makeother\"%
125 }

```

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

```

126 \def\MT@restore@catcodes#1{%
127   \ifx\relax#1\else
128     \noexpand\catcode'\noexpand#1\the\catcode'#1\relax
129     \expandafter\MT@restore@catcodes
130   \fi
131 }
132 \edef\MT@restore@catcodes{%
133   \MT@restore@catcodes\^-\=\*\,\/\\'\'\"\\relax
134 }

135 \MT@catcodes
136 \AtEndOfPackage{\MT@restore@catcodes}

```

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

```
137 \RequirePackage{keyval}[1997/11/10]
```

\MT@toks We need a token register.

```
138 \newtoks\MT@toks
```

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

```

\ifMT@expansion 139 \newif\ifMT@protrusion
\ifMT@auto       140 \newif\ifMT@expansion
\ifMT@selected  141 \newif\ifMT@auto
\ifMT@noligatures 142 \newif\ifMT@selected
\ifMT@draft      143 \newif\ifMT@noligatures
\ifMT@draft      144 \newif\ifMT@draft
\ifMT@spacing    145 <beta>
\ifMT@kerning    146 \newif\ifMT@spacing
\ifMT@babel      147 \newif\ifMT@kerning
\ifMT@babel      148 \newif\ifMT@babel
\ifMT@babel      149 </beta>

```

\MT@ifdocument Our private test whether we're still in the preamble

```
150 \newif\ifMT@document
```

\MT@pr@level ... and numbers.

```

\MT@pr@factor 151 \let\MT@pr@level\tw@
\MT@pr@unit   152 \let\MT@pr@factor\@m
\MT@ex@level  153 \let\MT@pr@unit\@empty
\MT@ex@level  154 \let\MT@ex@level\tw@
\MT@ex@factor 155 \let\MT@ex@factor\@m
\MT@stretch   156 \let\MT@stretch\m@ne
\MT@shrink    157 \let\MT@shrink \m@ne
\MT@shrink    158 \let\MT@step \m@ne
\MT@step      159 <beta>
\MT@sp@factor 160 \let\MT@sp@factor\@m
\MT@sp@unit    161 \let\MT@kn@factor\@m
\MT@kn@factor
\MT@kn@unit

```

Default unit for spacing settings is space, default unit for kerning is 1em.

```

162 \let\MT@sp@unit\m@ne
163 \def\MT@kn@unit{1em}
\MT@letterspacing 164 \let\MT@letterspacing\m@ne
165 </beta>

```

```

\MT@pr@min Minimum and maximum values allowed by pdfTEX.
\MT@pr@max 166 \def\MT@pr@min{-\@m}
\MT@ex@min 167 \let\MT@pr@max\@m
\MT@ex@max 168 \let\MT@ex@min\z@
\MT@ex@max 169 \let\MT@ex@max\@m
\MT@sp@min 170  $\beta$ 
\MT@sp@max 171 \def\MT@sp@min{-\@m}
\MT@sp@max 172 \let\MT@sp@max\@m
\MT@kn@min 173 \def\MT@kn@min{-\@m}
\MT@kn@max 174 \let\MT@kn@max\@m
\MT@kn@max 175  $\beta$ 

\MT@factor@default Default values for expansion.
\MT@stretch@default 176 \def\MT@factor@default{1000 }
\MT@shrink@default 177 \def\MT@stretch@default{20 }
\MT@step@default 178 \def\MT@shrink@default{20 }
\MT@step@default 179 \def\MT@step@default{4 }

\MT@letterspacing@default Default value for letterspacing (in thousandths of 1em).
\MT@letterspacing@default 180  $\beta$ \def\MT@letterspacing@default{100 }

```

## 13.2 Compatibility

For the record, the following L<sup>A</sup><sub>TEX</sub> commands will be modified by microtype:

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

```

\MT@pdfcprot@error Our competitor, the pdfcprot package, must not be tolerated!
181 \def\MT@pdfcprot@error{%
182   \MT@error{Detected the 'pdfcprot' package!\MessageBreak
183     '\MT@MT' and 'pdfcprot' may not be used together}{%
184     The 'pdfcprot' package provides an interface to character protrusion.\MessageBreak
185     So does the '\MT@MT' package. Using both packages at the same\MessageBreak
186     time will almost certainly lead to undesired results. Have your choice!}%
187   \let\MT@pdfcprot@error\relax
188 }
189 \ifpackageloaded{pdfcprot}\MT@pdfcprot@error\relax

\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 pdfTEX version 1.21b together with ledpatch.sty as of 2005/06/02 (v0.4),
character protrusion will work at last.

Peter Wilson was so kind to provide the \l@dunhbox@line hook in ledmac to
allow for protrusion. \leftmarginkern and \rightmarginkern are new primitives
of pdfTEX 1.21b (aka 1.30.0).
190 \def\MT@ledmac@setup{%
191   \ifMT@protrusion
192     \MT@requires@pdftex5{%

```

```

193 \MT@ifdefined@c@TF\l@dunhbox@line{%
194 \MT@info@nl{Patching ledmac to enable character protrusion}%
195 \newdimen\MT@led@kern
196 \let\MT@led@unhbox@line\l@dunhbox@line
197 \renewcommand*{\l@dunhbox@line}[1]{%
198 \ifhbox##1%
199 \MT@led@kern=\rightmarginkern##1%
200 \kern\leftmarginkern##1%
201 \MT@led@unhbox@line#1%
202 \kern\MT@led@kern
203 \fi
204 }%
205 }{%
206 \MT@warning@nl{%
207 Character protrusion in paragraphs with line\MessageBreak
208 numbering will only work if you update ledmac}%
209 }%
210 }{%
211 \MT@warning@nl{%
212 The pdftex version you are using does not allow\MessageBreak
213 character protrusion in paragraphs with line\MessageBreak
214 numbering by the 'ledmac' package.\MessageBreak
215 Upgrade pdftex to version 1.30 or later}%
216 }%
217 \fi
218 }

```

`\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`.)

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

The `chemsym` package redefines, among other commands, the Hungarian umlaut `\H` in a way that cannot be parsed by microtype. As a work-around, we restore the usual definition of `\H` before setting up the font (which will be done inside a group). — Since version 1.7, this is no longer needed, since our character parsing is robust enough now.

Same for `statex`. — No longer needed, either.

Spanish `babel` modifies the percent character, storing the original meaning in `\percentsign`.

```

220 \ifpackagewith{babel}{spanish}{%
221 \MT@ifdefined@c@T\percentsign{\let\%\percentsign}%
222 }\relax

```

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.)

```

223 \@ifpackageloaded{csquotes}{%
224 \@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax

```

```
225 } \relax
```

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

```
226 \ifpackageloaded{hyperref}{%
227   \chardef\%\%
228   \chardef\#\#
229 } \relax
230 }
```

`\MT@setup@microtype` 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.

```
231 \let\MT@setup@microtype\@empty
```

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

```
232 \def\MT@addto@setup#1{%
233   \begingroup
234     \MT@toks\expandafter{\MT@setup@microtype#1}%
235     \xdef\MT@setup@microtype{\the\MT@toks}%
236   \endgroup
237 }
```

It will be executed at the end of the preamble.

```
238 \AtBeginDocument{\MT@setup@microtype}
```

Check again at the end of the preamble.

```
239 \MT@addto@setup{%
240   \ifpackageloaded{pdfcpot}\MT@pdfcpot@error\relax
241   \ifpackageloaded{ledmac}\MT@ledmac@setup\relax
```

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

```
242   \let\MT@setupfont@hook\@empty
243   \ifpackagewith{babel}{spanish}{%
244     \g@addto@macro\MT@setupfont@hook{%
245       \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
246   } \relax
247   \ifpackageloaded{csquotes}{%
248     \ifpackagelater{csquotes}{2005/05/11}{%
249       \g@addto@macro\MT@setupfont@hook{\@disablequotes
250     }{%
251       \MT@warning@nl{%
252         Should you receive warnings about unknown slot\MessageBreak
253         numbers, try upgrading the 'csquotes' package}%
254     }%
255   } \relax
256   \ifpackageloaded{hyperref}{%
257     \g@addto@macro\MT@setupfont@hook{%
258       \chardef\%\%
259       \chardef\#\#
260     }%
```

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

```
261   \pdfstringdefDisableCommands{%
262     \let\pickup@font\MT@orig@pickupfont
263   } *beta
264   \let\lststyle\@empty
265   \def\textls#1#{\@firstofone}%
```

```

266 /beta
267 }%
268 }\relax
269 }

```

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

```
270 \expandafter\ifx\the\font\nullfont\normalfont\fi
```

### 13.3 Auxiliary macros

```

\MT@requires@etex For definitions that depend on e-TeX features.
271 \expandafter\let\expandafter\MT@requires@etex
272 \ifcase 0%
273   \ifx\TeXversion\undefined 1\else
274   \ifx\TeXversion\relax      1\else
275   \ifcase\TeXversion        1\fi
276   \fi
277 \fi\space
278 \@firstoftwo
279 \else
280 \@secondoftwo
281 \fi

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

\MT@edef@n Its expanding version.
283 \def\MT@edef@n#1{\expandafter\edef\csname #1\endcsname}

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

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

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

\MT@exp@string Remove trailing space.
287 \def\MT@exp@string{\expandafter\string}

\MT@exp@one@n Expand the second token once and enclose it in braces.
288 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}

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

\MT@exp@two@n Expand the next two tokens after <#1> once and enclose them in braces.
290 \def\MT@exp@two@n#1#2#3{%
291   \expandafter\expandafter\expandafter
292   #1\expandafter\expandafter\expandafter
293   {\expandafter#2\expandafter}\expandafter{#3}}

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

```



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

```

\MT@ifdefined@n@T 294 \MT@requires@etex{%
295 debug \MT@info@n1{0}{running etex}%
296 \def\MT@ifdefined@c@TF#1{%
297   \ifdefined#1%
298     \expandafter\@firstoftwo
299   \else
300     \expandafter\@secondoftwo
301   \fi
302 }
303 \def\MT@ifdefined@c@T#1{%
304   \ifdefined#1%
305     \expandafter\@firstofone
306   \else
307     \expandafter\@gobble
308   \fi
309 }
310 \def\MT@ifdefined@n@TF#1{%
311   \ifcsname#1\endcsname
312     \expandafter\@firstoftwo
313   \else
314     \expandafter\@secondoftwo
315   \fi
316 }
317 \def\MT@ifdefined@n@T#1{%
318   \ifcsname#1\endcsname
319     \expandafter\@firstofone
320   \else
321     \expandafter\@gobble
322   \fi
323 }
324 }{%
325 debug \MT@info@n1{0}{not running etex}%
326 \def\MT@ifdefined@c@TF#1{%
327   \ifx#1\@undefined
328     \expandafter\@secondoftwo
329   \else
330     \expandafter\@firstoftwo
331   \fi
332 }
333 \def\MT@ifdefined@c@T#1{%
334   \ifx#1\@undefined
335     \expandafter\@gobble
336   \else
337     \expandafter\@firstofone
338   \fi
339 }
340 \def\MT@ifdefined@n@TF#1{%
341   \begingroup\MT@exp@two@c\endgroup
342   \ifx\csname #1\endcsname\relax
343     \expandafter\@secondoftwo
344   \else
345     \expandafter\@firstoftwo
346   \fi
347 }
348 \def\MT@ifdefined@n@T#1{%
349   \begingroup\MT@exp@two@c\endgroup

```

```

350 \ifx\csname #1\endcsname\relax
351 \expandafter\@gobble
352 \else
353 \expandafter\@firstofone
354 \fi
355 }
356 }

\MT@ifempty Test whether argument is empty.
357 \begingroup
358 \catcode'\%=12
359 \catcode'\&=14
360 \gdef\MT@ifempty#1{&
361 \if %#1%&
362 \expandafter\@firstoftwo
363 \else
364 \expandafter\@secondoftwo
365 \fi
366 }
367 \endgroup

\MT@ifnumber Test whether argument is a number [0-9] (using an old trick by Mr. Arseneau).
368 \MT@requires@pdfTeX5{&
369 \def\MT@ifnumber#1{&
370 \ifcase\pdfmatch{^[0-9]+ *$}{#1}\relax
371 \expandafter\@secondoftwo
372 \else
373 \expandafter\@firstoftwo
374 \fi
375 }
376 }{&
377 \def\MT@ifnumber#1{&
378 \if!\ifnum9<1#1!\else?\fi
379 \expandafter\@firstoftwo
380 \else
381 \expandafter\@secondoftwo
382 \fi
383 }
384 }

\MT@ifdimen Test whether argument is dimension (or number).
385 \def\MT@ifdimen#1{&
386 \setbox\z@=\hbox{&
387 \MT@count=1#1\relax
388 \ifnum\MT@count=\@ne
389 \aftergroup\@secondoftwo
390 \else
391 \aftergroup\@firstoftwo
392 \fi}&
393 }

\MT@ifgt Test whether dimensions are smaller, larger or equal.
\MT@iflt
\MT@ifeq
394 \def\MT@ifgt#1#2{&
395 \ifdim #1\p@ > #2\p@
396 \expandafter\@firstoftwo
397 \else
398 \expandafter\@secondoftwo
399 \fi
400 }

```

```

401 \def\MT@iflt#1#2{%
402   \ifdim #1\p@ < #2\p@
403     \expandafter\@firstoftwo
404   \else
405     \expandafter\@secondoftwo
406   \fi
407 }
408 \def\MT@ifeq#1#2{%
409   \ifdim #1\p@ = #2\p@
410     \expandafter\@firstoftwo
411   \else
412     \expandafter\@secondoftwo
413   \fi
414 }

```

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

```

415 \MT@requires@pdftex5{%
416   \def\MT@ifstreq#1#2{%
417     \ifcase\pdfstrcmp{#1}{#2}\relax
418     \expandafter\@firstoftwo
419   \else
420     \expandafter\@secondoftwo
421   \fi
422 }
423 }%
424 \def\MT@ifstreq#1#2{%
425   \edef\x{#1}%
426   \edef\y{#2}%
427   \ifx\x\y
428     \expandafter\@firstoftwo
429   \else
430     \expandafter\@secondoftwo
431   \fi
432 }
433 }

```

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

```

434 \def\MT@xadd#1#2{%
435   \ifx#1\relax
436     \xdef#1{#2}%
437   \else
438     \xdef#1{#1#2}%
439   \fi
440 }

```

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

```

441 \def\MT@xaddb#1#2{%
442   \ifx#1\relax
443     \xdef#1{#2}%
444   \else
445     \xdef#1{#2#1}%
446   \fi
447 }

```

`\MT@map@clist@n` Run `<#2>` on all elements of the comma list `<#1>`. This and the following is mod-

`\MT@map@clist@c` modelled after `LATEX3` commands.

```

\MT@map@clist@c 448 \def\MT@map@clist@n#1#2{%
\MT@clist@function 449   \ifx\@empty#1\else
\MT@clist@break 450     \def\MT@clist@function##1{#2}%

```

```

451 \MT@map@clist@#1,\@nil,\@nnil
452 \fi
453 }
454 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
455 \def\MT@map@clist@#1,{%
456 \ifx\@nil#1%
457 \expandafter\MT@clist@break
458 \fi
459 \MT@clist@function{#1}%
460 \MT@map@clist@
461 }
462 \let\MT@clist@function\@gobble
463 \def\MT@clist@break#1\@nnil{}
```

`\MT@map@tlist@n` Execute `<#2>` on all elements of the token list `<#1>`. `\MT@tlist@break` can be used to jump out of the loop.

```

\MT@map@tlist@c 464 \def\MT@map@tlist@n#1#2{%
\MT@map@tlist@ 465 \MT@map@tlist@#2#1\@nnil
\MT@tlist@break 466 }
467 \def\MT@map@tlist@c#1#2{%
468 \expandafter\MT@map@tlist@
469 \expandafter#2#1\@nnil
470 }
471 \def\MT@map@tlist@#1#2{%
472 \ifx\@nnil#2\else
473 #1{#2}%
474 \expandafter\MT@map@tlist@
475 \expandafter#1%
476 \fi
477 }
478 \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 479 \newif\ifMT@inlist@
480 \def\MT@in@clist#1#2{%
481 \def\x##1,#1,##2##3\@nnil{%
482 \ifx##2\@empty
483 \MT@inlist@false
484 \else
485 \MT@inlist@true
486 \fi
487 }%
488 \expandafter\x\expandafter,#2,#1,\@empty\@nnil
489 }
```

`\MT@rem@from@clist` Remove item `<#1>` from comma list `<#2>`. Using `\pdfmatch` and `\pdflastmatch` here would be really slow!

```

490 \def\MT@rem@from@clist#1#2{%
491 \def\x##1,#1,##2##3\@nnil{%
492 \ifx##2\@empty\else
493 \ifx\##1\global\let#2\@gobble
494 \else\xdef#2{\@gobble#1}\fi
495 \def\x###1,#1,###2###3\@nnil{%
496 \xdef#2{#2###1}%
497 }%
498 \x,##2##3,#1,\@nnil
499 \fi
500 }%
501 \expandafter\x\expandafter,#2,#1,\@empty\@nnil
```

```

502 }

\MT@in@tlist Test whether item is in token list. Since this isn't too elegant, I thought that at
\MT@in@tlist@ least here, \pdfmatch would be more efficient – however, it turned out to be even
                slower than this solution.
503 \def\MT@in@tlist#1#2{%
504   \MT@inlist@false
505   \def\x{#1}%
506   \MT@map@tlist@c#2\MT@in@tlist@
507 }
508 \def\MT@in@tlist@#1{%
509   \edef\y{#1}%
510   \ifx\x\y
511     \MT@inlist@true
512     \expandafter\MT@tlist@break
513   \fi
514 }

\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@@ 515 \def\MT@in@rlist#1{%
\MT@size@name 516   \MT@inlist@false
517   \MT@map@tlist@c#1\MT@in@rlist@
518 }
519 \def\MT@in@rlist@#1{%
520   \expandafter\MT@in@rlist@@#1%
521 }
522 \def\MT@in@rlist@@#1#2#3{%
523   \MT@ifeq{#2}\m@ne{%
524     \MT@ifeq{#1}\MT@size
525     \MT@inlist@true
526     \relax
527   }{%
528     \MT@iflt\MT@size{#1}\relax{%
529       \MT@iflt\MT@size{#2}%
530       \MT@inlist@true
531       \relax
532     }%
533   }%
534   \ifMT@inlist@
535     \def\MT@size@name{#3}%
536     \expandafter\MT@tlist@break
537   \fi
538 }

\MT@loop This is the same as LATEX's \loop, which we mustn't use, since this could confuse
\MT@iterate an outer \loop in the document.
\MT@repeat 539 \def\MT@loop#1\MT@repeat{%
540   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
541   \MT@iterate
542   \let\MT@iterate\relax
543 }
544 \let\MT@repeat\fi

\MT@do@font Execute <#1> 255 times.
545 \def\MT@do@font#1{%
546   \@tempcnta\z@
547   \MT@loop #1%

```

```

548 \advance\@tempcnta \@ne
549 \ifnum\@tempcnta < \@ccclvi \MT@repeat
550 }

```

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

```

\MT@increment
551 \newcount\MT@count
552 \MT@requires@etex{%
553 \def\MT@increment#1{%
554 \edef#1{\number\numexpr #1 + 1\relax}%
555 }
556 }{%
557 \def\MT@increment#1{%
558 \MT@count=#1\relax
559 \advance\MT@count \@ne
560 \edef#1{\number\MT@count}%
561 }
562 }

```

`\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.

```

563 \MT@requires@etex{%
564 \def\MT@scale#1#2#3{%
565 \ifnum #3 = \z@
566 #1=\numexpr #1 * #2\relax
567 \else
568 #1=\numexpr #1 * #2 / #3\relax
569 \fi
570 }
571 }{%
572 \def\MT@scale#1#2#3{%
573 \multiply #1 #2\relax
574 \ifnum #3 = \z@ \else
575 \divide #1 #3\relax
576 \fi
577 }
578 }

```

`\MT@make@string` Set the category code of all characters to 12.

```

579 \let\MT@make@string\@onelevel@sanitize

```

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

```

\MT@abbr@ex
\MT@abbr@pr@c 580 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 581 \def\MT@abbr@ex{expansion}
582 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@pr@inh 583 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh 584 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@n1 585 \def\MT@abbr@ex@inh{expansion inheritance}
\MT@abbr@sp 586 \def\MT@abbr@n1{noligatures}
587 *beta
\MT@abbr@sp@c 588 \def\MT@abbr@sp{spacing}
589 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@sp@inh 590 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@kn 591 \def\MT@abbr@kn{kerning}
\MT@abbr@kn@c 592 \def\MT@abbr@kn@c{kerning codes}
\MT@abbr@kn@inh

```

```

593 \def\MT@abbr@kn@inh{kerling inheritance}
594 </beta>
\MT@rbba@protrusion These we also need the other way round.
\MT@rbba@expansion 595 \def\MT@rbba@protrusion{pr}
\MT@rbba@spacing 596 \def\MT@rbba@expansion{ex}
\MT@rbba@kerning 597 <*beta>
598 \def\MT@rbba@spacing{sp}
599 \def\MT@rbba@kerning{kn}
600 </beta>

```

### 13.4 Setting up a font

\MT@setupfont Setting up a font entails checking whether protrusion/expansion is desired for the current font (\MT@font@name), and if so, adjusting \lcode and \rcode (protrusion) and \efcode (expansion) for each character.

```

601 \def\MT@setupfont{%
602   \ifx\MT@vinfo\MT@info\nl
603     \MT@info{Setting up font '\MT@exp@string\MT@font'}\fi

```

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

```
604 \MT@setupfont@hook
```

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

```
605 \MT@exp@two@c\MT@split@name\string\MT@font\nl
```

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

```

606 \MT@exp@one@n\MT@find@file\MT@family
607 \ifx\MT@familyalias\@empty \else
608   \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.)

```
609 \ifx\f@encoding\cf@encoding\else\@enc@update\fi
```

Now we can begin setting up the font for all features. The following commands are \let to \relax if the respective feature is generally disabled.

Protrusion has to be set up first, says Thành!

```

610 \MT@protrusion
611 \MT@expansion

```

Interword spacing and kerning.

```

612 <*beta>
613 \MT@spacing
614 \MT@kerning
615 </beta>

```

Disable ligatures?

```

616 \MT@noligatures
617 }

```

`\MT@split@name` Split up the font name.

```
\MT@encoding 618 \def\MT@split@name#1/#2/#3/#4/#5\@nil{%
\MT@family    619 \def\MT@encoding{#1}%
\MT@series    620 \def\MT@family{#2}%
\MT@shape     621 \def\MT@series{#3}%
\MT@size      622 \def\MT@shape{#4}%
\MT@size      623 \def\MT@size{#5}%
```

`\MT@familyalias` Alias family?

```
624 \MT@ifdefined@n@TF{MT@\MT@family @alias}%
625 {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
626 {\let\MT@familyalias\empty}%
627 }
```

`\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@maybe@do 628 \newif\ifMT@do
629 \def\MT@maybe@do#1{%
```

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

```
630 \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).

```
631 \MT@dotrue
632 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
633 \MT@ifdefined@n@TF{MT@checklist@##1}%
634 {\csname MT@checklist@##1\endcsname}%
635 {\MT@checklist@{##1}}%
636 {#1}%
637 }%
638 \else
639 \MT@dofalse
640 \fi
641 \ifMT@do
```

`\MT@feat` stores the current feature.

```
642 \def\MT@feat{#1}%
643 \csname MT@set@#1@codes\endcsname
644 \else
645 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
646 \fi
647 }
```

`\MT@checklist@` The generic test.

```
648 \def\MT@checklist@#1#2{%
649 \edef\@tempa{\csname MT@#2@setname\endcsname}%
650 \!debug \MT@ifdefined@n@T
651 \!debug \MT@ifdefined@n@TF
652 {MT@#2list@#1@\@tempa}{%
```

Begin a `\expandafter` orgy to test whether the font characteristic is in the list.

```
653 \expandafter\expandafter\expandafter
654 \MT@in@clist\expandafter\expandafter\expandafter
655 {\csname MT@#1\expandafter\endcsname\expandafter}%
656 \csname MT@#2list@#1@\@tempa\endcsname
657 \ifMT@inlist@
```



```

658 <debug>\MT@info@n1{1}{\@nameuse{MT@abbr@#2}: #1 '\@nameuse{MT@#1}' in list}%
659 \MT@dotrue
660 \else
661 <debug>\MT@info@n1{1}{\@nameuse{MT@abbr@#2}: #1 '\@nameuse{MT@#1}' not in list}%
662 \MT@dofalse
663 \expandafter\MT@clist@break
664 \fi
665 }%

```

If no limitations have been specified, i.e. the list for a font characteristic has not been defined at all, the font should be expanded resp. protruded.

```

666 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#2}: #1 list empty}}%
667 }

```

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

```

668 \def\MT@checklist@font#1{%
669 \edef\@tempa{\csname MT@#1@setname\endcsname}%
670 <!debug> \MT@ifdefined@n@T
671 <debug> \MT@ifdefined@n@TF
672 {MT@#1list@font@\@tempa}{%

```

There mustn't be a space after the font name, hence we have to stringify it. There surely is a better way than this silly chain, however, I'm beginning to be haunted by \expandafters in my dreams, so I have to leave it at that.

```

673 \expandafter\expandafter\expandafter\MT@exp@one@n
674 \expandafter\expandafter\expandafter\MT@in@clist
675 \expandafter\expandafter\expandafter{%
676 \expandafter\expandafter\expandafter\string
677 \expandafter\MT@font\expandafter}%
678 \csname MT@#1list@font@\@tempa\endcsname
679 \ifMT@inlist@
680 <debug>\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: font '\MT@font' in list}%
681 \expandafter\MT@clist@break
682 \else
683 <debug>\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: font '\MT@font' not in list}%
684 \MT@dofalse
685 \fi
686 }%
687 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: font list empty}}%
688 }

```

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

```

689 \def\MT@checklist@family#1{%
690 \edef\@tempa{\csname MT@#1@setname\endcsname}%
691 <!debug> \MT@ifdefined@n@T
692 <debug> \MT@ifdefined@n@TF
693 {MT@#1list@family@\@tempa}{%
694 \MT@exp@two@n\MT@in@clist
695 \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
696 \ifMT@inlist@
697 <debug>\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family '\@nameuse{MT@family}' in list}%
698 \MT@dotrue
699 \else
700 <debug>\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family '\@nameuse{MT@family}' not in list}%
701 \MT@dofalse
702 \ifx\MT@familyalias\@empty \else
703 \MT@exp@two@n\MT@in@clist
704 \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
705 \ifMT@inlist@

```

```

706 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: alias '\MT@familyalias' in list}%
707 \MT@dotrue
708 <debug> \else\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: alias '\MT@familyalias' not in list}%
709 \fi
710 \fi
711 \fi
712 \ifMT@do \else
713 \expandafter\MT@clist@break
714 \fi
715 }%
716 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family list empty}}%
717 }

```

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

```

718 \def\MT@checklist@size#1{%
719 \edef\@tempa{\csname MT@#1@setname\endcsname}%
720 <!debug> \MT@ifdefined@n@T
721 <debug> \MT@ifdefined@n@TF
722 {\MT@#1list@size@\@tempa}%
723 \expandafter\MT@in@rlist
724 \csname MT@#1list@size@\@tempa\endcsname
725 \ifMT@inlist@
726 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size '\MT@size' in list}%
727 \MT@dotrue
728 \else
729 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size '\MT@size' not in list}%
730 \MT@dofalse
731 \expandafter\MT@clist@break
732 \fi
733 }%
734 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size list empty}}%
735 }

```

### 13.4.1 Protrusion

**\MT@protrusion** Set up for protrusion?

```
736 \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.

```

737 \def\MT@set@pr@codes{%
738 \MT@reset@pr@codes

```

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

```

739 \MT@if@list@exists{%
740 \MT@get@dimen@six
741 \MT@get@opt

```

Get the name of the inheritance list and parse it.

```
742 \MT@get@inh@list
```

Load additional lists?

```
743 \MT@load@list{\MT@pr@c@name}%
```

Load the main list.

```

744 \edef\MT@curr@list@name{protrusion list '\MT@pr@c@name'}%
745 \MT@let@cn\@tempc{MT@pr@c@\MT@pr@c@name}%
746 \expandafter\MT@pr@do\@tempc,\relax,%

```

```

747 } \relax
748 }

\MT@set@all@pr Set all protrusion codes of the font.
749 \def\MT@set@all@pr#1#2{%
750 (debug) \MT@info@n1{3}{-- lp/rp: setting all to \number#1/\number#2}%
751 \MT@do@font{%
752 \lcode\MT@font\@tempcnta=#1\relax
753 \rcode\MT@font\@tempcnta=#2\relax
754 }%
755 }

\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.
756 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
757 \let\MT@reset@pr@codes\relax

\MT@gobble@settings If \fontdimen 6 is zero, character protrusion won't work, and we can skip the set-
\MT@dimen@six tings (for example, the dsfont fonts don't specify this dimension; this is probably
\MT@get@dimen@six a bug).
758 \def\MT@get@dimen@six{%
759 \ifnum\fontdimen6\MT@font=\z@
760 \MT@warning@n1{%
761 Font '\MT@exp@string\MT@font' does not specify its\MessageBreak
762 \backslashchar fontdimen 6 (width of an 'em')! Therefore,\MessageBreak
763 \nameuse{MT@abbr@MT@feat} will not work with this font}%
764 \expandafter\MT@gobble@settings
765 \else
766 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
767 \fi
768 }
769 \def\MT@gobble@settings#1\@tempc,\relax,{}

\MT@pr@do Split up the values and set \lcode and \rcode.
770 \def\MT@pr@do#1,{%
771 \ifx\relax#1\@empty\else
772 \MT@pr@split #1=\relax
773 \expandafter\MT@pr@do
774 \fi
775 }

\MT@pr@split The keyval package would remove spaces here, which we needn't do since
\SetProtrusion ignores spaces in the protrusion list anyway.
776 \def\MT@pr@split#1=#2=#3\relax{%
777 \def\@tempa{#1}%
778 \ifx\@tempa\@empty \else
779 \MT@get@slot
780 \ifnum\MT@char > \m@ne
781 \MT@get@char@unit
782 \MT@pr@split@val#2\relax
783 \fi
784 \fi
785 }

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

```

```

788 \MT@ifempty\@tempb\relax{%
789 \MT@scale@to@em
790 \lcode\MT@font\MT@char=\@tempcntb
791 (debug)\MT@info@n{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char: [#1]}%
792 }%
793 \def\@tempb{#2}%
794 \MT@ifempty\@tempb\relax{%
795 \MT@scale@to@em
796 \rcode\MT@font\MT@char=\@tempcntb
797 (debug)\MT@info@n{4}{;;; rp (\MT@char): \number\rcode\MT@font\MT@char: [#2]}%
798 }%

```

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

```

799 \MT@ifdefined@c@T\MT@pr@inh@name{%
800 \MT@ifdefined@n@T\MT@inh@MT@pr@inh@name @\MT@char @}{%
801 \expandafter\MT@map@tlist@c
802 \csname MT@inh@MT@pr@inh@name @\MT@char @\endcsname
803 \MT@set@pr@heirs
804 }%
805 }%
806 }

```

`\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 `\[lr]pcode`’s limit is 1000). Now, the maximum protrusion is 1em of the font.

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

```

807 \MT@requires@pdfTeX\thr@@{%
808 \def\MT@scale@to@em{%
809 \@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`.

```

810 \MT@scale\@tempcntb \@tempb \MT@dimen@six
811 \ifnum\@tempcntb=\z@ \else
812 \MT@scale@factor
813 \fi
814 }

```

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

```

815 \MT@requires@etex{%
816 \def\MT@get@charwd{%
817 \MT@count=\number\fontcharwd\MT@font\MT@char\relax
818 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
819 }
820 }{%
821 \def\MT@get@charwd{%

```

```

822     \setbox\z@=\hbox{\MT@font \char\MT@char}%
823     \MT@count=\wd\z@\relax
824     \ifnum\MT@count=\z@ \MT@info@missing@char \fi
825   }
826 }

```

No adjustment with versions 0.14f and 0.14g.

```

827 }{%
828   \def\MT@scale@to@em{%
829     \MT@count=\@tempb\relax
830     \ifnum\MT@count=\z@ \else
831       \MT@scale@factor
832     \fi
833   }

```

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

```

834   \def\MT@get@charwd{\MT@count=\MT@dimen@six}
835 }

```

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

```

836 \def\MT@get@font@dimen#1{%
837   \MT@count=\number\fontdimen#1\MT@font
838 }

```

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

```

839 \MT@requires@etex{%
840   \def\MT@info@missing@char{%
841     \MT@info@n1{Character '\the\MT@toks' \iffontchar\MT@font\MT@char
842       has a width of 0pt \else is missing \fi in font\MessageBreak
843       '\MT@exp@string\MT@font'. Ignoring protrusion settings\MessageBreak
844       for this character}%
845   }
846 }{%
847   \def\MT@info@missing@char{%
848     \MT@info@n1{%
849       Character '\the\MT@toks' has a width of 0pt\MessageBreak
850       (it's probably missing) in font '\MT@exp@string\MT@font'.\MessageBreak
851       Ignoring protrusion settings for this character}%
852   }
853 }

```

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

```

854 \def\MT@scale@factor{%
855   \ifnum\csname MT@\MT@feat @factor@endcsname=\@m \else
856     \expandafter\MT@scale\expandafter
857     \@tempcntb \csname MT@\MT@feat @factor@endcsname \@m
858   \fi
859   \ifnum\@tempcntb>\csname MT@\MT@feat @max@endcsname\relax
860     \@tempcnta=\csname MT@\MT@feat @max@endcsname
861     \MT@warn@code@too@large
862   \else
863     \ifnum\@tempcntb<\csname MT@\MT@feat @min@endcsname\relax
864       \@tempcnta=\csname MT@\MT@feat @min@endcsname
865       \MT@warn@code@too@large
866     \fi
867   \fi
868 }

```

\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 file.

```

869 \def\MT@warn@code@too@large{%
870   \ifnum\csname MT@\MT@feat @factor@endcsname=\@m \else
871     \expandafter\MT@scale\expandafter\@tempcnta\expandafter\@m
872       \csname MT@\MT@feat @factor@endcsname
873   \fi
874   \MT@scale\@tempcnta \MT@dimen@six \MT@count
875   \MT@warning@n1{The \nameuse{MT@abbr@\MT@feat} code \@tempb\space
876     is too large for character\MessageBreak
877     'the\MT@toks' in \MT@curr@list@name.\MessageBreak
878     Setting it to the maximum of \number\@tempcnta}%
879   \@tempcntb=\@tempcnta
880 }
```

**\MT@get@opt** The optional argument to `\SetProtrusion`, `\SetExtraSpacing` and `\SetExtraKerning` (`\SetExpansion` is being dealt with in `\MT@get@ex@opt`).

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

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

```

\MT@sp@factor@ 882 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name@endcsname @factor}{%
\MT@kn@factor@ 883   \MT@let@nn{MT@\MT@feat @factor@}
884     {MT@\MT@feat @c@\csname MT@\MT@feat @c@name@endcsname @factor}%
885     \MT@vinfo{... : Multiplying \nameuse{MT@abbr@\MT@feat} codes by
886               \number\csname MT@\MT@feat @factor@endcsname/1000}%
887   }{%
888     \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
889   }%
```

**\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@ 890 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name@endcsname @unit}{%
891   \MT@let@nn{MT@\MT@feat @unit@}%
892   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name@endcsname @unit}%
893   \expandafter\ifx\csname MT@\MT@feat @unit@endcsname\@empty
894     \MT@vinfo{... : Setting \nameuse{MT@abbr@\MT@feat} codes
895               relative to character widths}%
896   \else
897     \expandafter\ifx\csname MT@\MT@feat @unit@endcsname\m@ne
898     \MT@vinfo{... : Setting \nameuse{MT@abbr@\MT@feat} codes
899               relative to width of space}%
900   \fi
901   \fi
902 }{%
903   \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
904 }%
```

**\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.

```

905 \let\MT@get@char@unit\relax
906 \let\MT@get@space@unit@gobble
907 \expandafter\ifx\csname MT@\MT@feat @unit@endcsname\@empty
908   \let\MT@get@char@unit\MT@get@charwd
909 \else
910   \expandafter\ifx\csname MT@\MT@feat @unit@endcsname\m@ne
911     \let\MT@get@space@unit\MT@get@font@dimen
912   \else
913     \expandafter\MT@get@unit\csname MT@\MT@feat @unit@endcsname
```

```
914 \fi
915 \fi
```

Preset all characters?

```
916 \MT@ifdefined@nT{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @preset}{%
917 \csname MT@preset@MT@feat\endcsname
918 }%
919 }
```

\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.

```
920 \def\MT@get@unit#1{%
921 \expandafter\MT@get@unit@#1 e!\@nil
922 \ifx\x\@empty\else\let#1x\fi
923 \@defaultunits\@tempdima#1 pt\relax\@nnil
924 \ifdim\@tempdima=\z@
925 \MT@warning@n1{%
926 Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
927 width. Setting factors of list '\@nameuse{MT@MT@feat @c@name}'\MessageBreak
928 relative to character widths instead}%
929 \let#1\@empty
930 \let\MT@get@char@unit\MT@get@charwd
931 \else
932 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
933 to \the\@tempdima}%
934 \MT@count=\number\@tempdima\relax
935 \fi
936 }
937 \def\MT@get@unit@#1e#2#3\@nil{%
938 \ifx\#3\@empty\else
939 \if m#2%
940 \edef\x{#1\fontdimen6\MT@font}%
941 \else
942 \if x#2%
943 \edef\x{#1\fontdimen5\MT@font}%
944 \fi
945 \fi
946 \fi
947 }
```

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

```
948 \def\MT@set@pr@heirs#1{%
949 \lcode\MT@font#1=\lcode\MT@font\MT@char
950 \rcode\MT@font#1=\rcode\MT@font\MT@char
951 <debug>
952 \MT@dinfo{n1}{-- heir of \MT@char: #1}%
953 \MT@dinfo{n4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
954 \number\rcode\MT@font\MT@char}%
955 </debug>
956 }
```

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

```
\MT@preset@pr 957 \def\MT@preset@pr{%
958 \expandafter\expandafter\expandafter\MT@preset@pr@
959 \csname MT@pr@c@MT@pr@c@name @preset\endcsname\@nil
960 }
961 \def\MT@preset@pr@#1,#2\@nil{%
```

```

962 \ifx\MT@pr@unit@\empty
963 \MT@warn@preset@tewidth{pr}%
964 \let\MT@preset@aux\MT@preset@aux@factor
965 \else
966 \let\MT@preset@aux\MT@preset@aux@space
967 \fi
968 \MT@preset@aux{#1}\@tempa
969 \MT@preset@aux{#2}\@tempb
970 \MT@set@all@pr@\@tempa\@tempb
971 }

```

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

```

\MT@preset@aux@factor 972 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 973 \@tempcntb=#1\relax
974 \MT@scale@factor
975 \edef#2{\number\@tempcntb}%
976 }
977 \def\MT@preset@aux@space#1#2{%
978 \def\@tempb{#1}%
979 \MT@get@space@unit\tw@
980 \MT@scale@to@em
981 \edef#2{\number\@tempcntb}%
982 }

```

\MT@warn@preset@tewidth

```

983 \def\MT@warn@preset@tewidth#1{%
984 \MT@warning@nl{%
985 \cannot preset characters relative to their widths\MessageBreak
986 for \@nameuse{MT@abbr@#1} list '\@nameuse{MT@#1@name}'. Presetting them%
987 \MessageBreak relative to lem instead}%
988 }

```

### 13.4.2 Expansion

\MT@expansion Set up for expansion?

```

989 \def\MT@expansion{\MT@maybe@do{ex}}

```

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

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

```

990 \def\MT@set@ex@codes@{%
991 \MT@if@list@exists{%
992 \MT@get@ex@opt
993 \MT@reset@ef@codes
994 \MT@get@inh@list
995 \MT@load@list{\MT@ex@name}%
996 \edef\MT@curr@list@name{expansion list '\MT@ex@name'}%
997 \MT@let@cn\@tempc{\MT@ex@code\MT@ex@code}%
998 \expandafter\MT@ex@do\@tempc,\relax,%
999 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1000 }\relax
1001 }

```

\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.

```

1002 \newif\ifMT@nonselected
1003 \def\MT@set@ex@codes@n{%
1004   \MT@nonselectedtrue
1005   \MT@if@list@exists
1006   \MT@get@ex@opt
1007   {%
1008     \let\MT@stretch@\MT@stretch
1009     \let\MT@shrink@\MT@shrink
1010     \let\MT@step@\MT@step
1011     \let\MT@auto@\MT@auto
1012     \let\MT@ex@factor@\MT@ex@factor
1013   }%
1014   \MT@reset@ef@codes
1015   \pdfontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1016   \MT@nonselectedfalse
1017 }
```

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

```
1018 \let\MT@set@ex@codes\MT@set@ex@codes@n
```

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

```

1019 \def\MT@set@all@ex#1{%
1020   (debug) \MT@dinfo@n{3}{-- ex: setting all to \number#1}%
1021   \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1022 }
1023 \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.

```

1024 \MT@requires@pdfTeX4{%
1025   \def\MT@reset@ef@codes{%
1026     \ifnum\MT@ex@factor@=\@m \else
1027       \MT@reset@ef@codes@
1028     \fi
1029   }
1030 }%
1031 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1032 }
```

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

```

1033 \def\MT@ex@do#1,{%
1034   \ifx\relax#1\@empty \else
1035     \MT@ex@split #1=\relax
1036     \expandafter\MT@ex@do
1037   \fi
1038 }
```

`\MT@ex@split`

```

1039 \def\MT@ex@split#1=#2=#3\relax{%
1040   \def\@tempa{#1}%
1041   \ifx\@tempa\@empty \else
1042     \MT@get@slot
1043     \ifnum\MT@char > \m@ne
1044       \@tempcntb=#2\relax
```

Take an optional factor into account.

```

1045     \ifnum\MT@ex@factor@=\@m \else
1046     \MT@scale\@tempcntb \MT@ex@factor@ \@m
1047     \fi
1048     \ifnum\@tempcntb > \MT@ex@max
1049     \MT@warn@ex@too@large\MT@ex@max
1050     \else
1051     \ifnum\@tempcntb < \MT@ex@min
1052     \MT@warn@ex@too@large\MT@ex@min
1053     \fi
1054     \fi
1055     \efcode\MT@font\MT@char=\@tempcntb
1056 (debug) \MT@edinfo\{4\}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#2]}%

```

Heirs, heirs, I love thy heirs.

```

1057     \MT@ifdefined@c@T\MT@ex@inh@name{%
1058     \MT@ifdefined@n@T{MT@inh\MT@ex@inh@name @\MT@char @}{%
1059     \expandafter\MT@map@tlist@c
1060     \csname MT@inh\MT@ex@inh@name @\MT@char @\endcsname
1061     \MT@set@ex@heirs
1062     }%
1063     }%
1064     \fi
1065     \fi
1066 }

```

\MT@warn@ex@too@large

```

1067 \def\MT@warn@ex@too@large#1{%
1068 \MT@warning@n{Expansion factor \number\@tempcntb\space too large for
1069 character\MessageBreak 'the\MT@toks' in \MT@curr@list@name.\MessageBreak
1070 Setting it to the maximum of \number#1}%
1071 \@tempcntb=#1\relax
1072 }

```

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

```

\MT@ex@factor@ 1073 \def\MT@get@ex@opt{%
\MT@stretch@ 1074 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1075 \MT@let@cn\MT@ex@factor@{MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1076 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 1077 }{%
1078 \let\MT@ex@factor@\MT@ex@factor
1079 }%
1080 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @stretch}{%
1081 \MT@let@cn\MT@stretch@{MT@ex@c@\MT@ex@c@name @stretch}%
1082 \MT@vinfo{... : Setting stretch limit to \number\MT@stretch@}%
1083 }{%
1084 \let\MT@stretch@\MT@stretch
1085 }%
1086 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @shrink}{%
1087 \MT@let@cn\MT@shrink@{MT@ex@c@\MT@ex@c@name @shrink}%
1088 \MT@vinfo{... : Setting shrink limit to \number\MT@shrink@}%
1089 }{%
1090 \let\MT@shrink@\MT@shrink
1091 }%
1092 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @step}{%
1093 \MT@let@cn\MT@step@{MT@ex@c@\MT@ex@c@name @step}%
1094 \MT@vinfo{... : Setting expansion step to \number\MT@step@}%
1095 }{%
1096 \let\MT@step@\MT@step
1097 }%
1098 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @auto}{%

```

```

1099 \MT@let@cn\MT@auto@\MT@ex@cc@\MT@ex@cc@name @auto}%
1100 \def\@tempa{autoexpand}%
1101 \MT@vinfo{... : \ifx\@tempa\MT@auto@ En\else Dis\fi
1102 abling automatic expansion}%
1103 }{%
1104 \let\MT@auto@\MT@auto
1105 }%
1106 }

\MT@set@ex@heirs
1107 \def\MT@set@ex@heirs#1{%
1108 \efcode\MT@font#1=\efcode\MT@font\MT@char
1109 <*debug>
1110 \MT@dinfo{n1}{2}{-- heir of \MT@char: #1}%
1111 \MT@dinfo{n1}{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1112 </debug>
1113 }

\MT@preset@ex
1114 \def\MT@preset@ex{%
1115 \@tempcntb=\csname MT@ex@cc@\MT@ex@cc@name @preset\endcsname\relax
1116 \MT@scale@factor
1117 \MT@set@all@ex@\tempcntb
1118 }
1119 <*beta>

```

### 13.4.3 Interword Space (Glue)

\MT@spacing Adjustment of interword spacing? Only for sufficiently new versions of pdf<sub>T</sub><sub>E</sub>X.

```

1120 \MT@requires@pdftex6{%
1121 \def\MT@spacing{\MT@maybe@do{sp}}
1122 }{%
1123 \let\MT@spacing\relax
1124 }

\MT@set@sp@codes This is all the same.
1125 \def\MT@set@sp@codes{%
1126 \MT@reset@sp@codes
1127 \MT@if@list@exists{%
1128 \MT@get@dimen@six
1129 \MT@get@opt
1130 \MT@get@inh@list
1131 \MT@load@list{\MT@sp@cc@name}%
1132 \edef\MT@curr@list@name{spacing list '\MT@sp@cc@name'}%
1133 \MT@let@cn\@tempc{\MT@sp@cc@\MT@sp@cc@name}%
1134 \expandafter\MT@sp@do\@tempc,\relax,%
1135 }\relax
1136 }

\MT@sp@do
1137 \def\MT@sp@do#1,{%
1138 \ifx\relax#1\@empty \else
1139 \MT@sp@split #1=\relax
1140 \expandafter\MT@sp@do
1141 \fi
1142 }

```

\MT@sp@split

```

1143 \def\MT@sp@split#1=#2=#3\relax{%
1144   \def\@tempa{#1}%
1145   \ifx\@tempa\empty \else
1146     \MT@get@slot
1147     \ifnum\MT@char > \m@ne
1148       \MT@get@char@unit
1149       \MT@sp@split@val#2\relax
1150     \fi
1151   \fi
1152 }

```

\MT@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).

```

1153 \def\MT@sp@split@val#1,#2,#3\relax{%
1154   \def\@tempb{#1}%
1155   \MT@ifempty\@tempb\relax{%
1156     \MT@get@space@unit\tw@
1157     \MT@scale@to@em
1158     \knbscode\MT@font\MT@char=\@tempcntb
1159     <debug>\MT@info@n1{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1160   }%
1161   \def\@tempb{#2}%
1162   \MT@ifempty\@tempb\relax{%
1163     \MT@get@space@unit\thr@@
1164     \MT@scale@to@em
1165     \stbscode\MT@font\MT@char=\@tempcntb
1166     <debug>\MT@info@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1167   }%
1168   \def\@tempb{#3}%
1169   \MT@ifempty\@tempb\relax{%
1170     \MT@get@space@unit4%
1171     \MT@scale@to@em
1172     \shbscode\MT@font\MT@char=\@tempcntb
1173     <debug>\MT@info@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1174   }%
1175   \MT@ifdefined@c@T\MT@sp@inh@name{%
1176     \MT@ifdefined@n@T\MT@inh@\MT@sp@inh@name @\MT@char @{%
1177       \expandafter\MT@map@tlist@c
1178       \csname MT@inh@\MT@sp@inh@name @\MT@char @\endcsname
1179       \MT@set@sp@heirs
1180     }%
1181   }%
1182 }

```

\MT@set@sp@heirs

```

1183 \def\MT@set@sp@heirs#1{%
1184   \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1185   \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1186   \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1187   <*debug>
1188   \MT@info@n1{2}{-- heir of \MT@char: #1}%
1189   \MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1190                                           \number\stbscode\MT@font\MT@char/%
1191                                           \number\shbscode\MT@font\MT@char}%
1192   </debug>
1193 }

```

\MT@set@all@sp

\MT@reset@sp@codes

\MT@reset@sp@codes@

```

1194 \def\MT@set@all@sp#1#2#3{%
1195 <debug>\MT@info@n1{3}{-- knbs/stbs/shbs: setting all to \number#1/\number#2/\number#3}%
1196 \MT@do@font{%
1197 \knbscode\MT@font\@tempcnta=#1\relax
1198 \stbscode\MT@font\@tempcnta=#2\relax
1199 \shbscode\MT@font\@tempcnta=#3\relax
1200 }%
1201 }
1202 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1203 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1204 \def\MT@preset@sp{%
1205 \expandafter\expandafter\expandafter\MT@preset@sp@
1206 \cscname MT@sp@cc@MT@sp@cc@name @preset\endcscname\@nil
1207 }
1208 \def\MT@preset@sp@#1,#2,#3\@nil{%
1209 \ifx\MT@sp@unit@empty
1210 \MT@warn@preset@twidth{sp}%
1211 \MT@preset@aux@factor{#1}\@tempa
1212 \MT@preset@aux@factor{#2}\@tempc
1213 \MT@preset@aux@factor{#3}\@tempb
1214 \else
1215 \MT@preset@aux@space{#1}\@tempa
1216 \def\@tempb{#2}%
1217 \MT@get@space@unit\thr@@
1218 \MT@scale@to@em
1219 \edef\@tempc{\number\@tempcntb}%
1220 \def\@tempb{#3}%
1221 \MT@get@space@unit4%
1222 \MT@scale@to@em
1223 \edef\@tempb{\number\@tempcntb}%
1224 \fi
1225 \MT@set@all@sp\@tempa\@tempc\@tempb
1226 }

```

### 13.4.4 Additional Kerning

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

```

1227 \MT@requires@pdftex6{%
1228 \def\MT@kerning{\MT@maybe@do{kn}}
1229 }%
1230 \let\MT@kerning\relax
1231 }

\MT@set@kn@codes
1232 \def\MT@set@kn@codes{%
1233 \MT@reset@kn@codes
1234 \MT@if@list@exists{%
1235 \MT@get@dimen@six
1236 \MT@get@opt
1237 \MT@get@inh@list
1238 \MT@load@list{\MT@kn@cc@name}%
1239 \edef\MT@curr@list@name{kerning list '\MT@kn@cc@name'}%
1240 \MT@let@cn\@tempc{MT@kn@cc@MT@kn@cc@name}%
1241 \expandafter\MT@kn@do\@tempc,\relax,%
1242 }\relax
1243 }

```

\MT@kn@do

```
1244 \def\MT@kn@do#1,{%
1245   \ifx\relax#1\@empty \else
1246     \MT@kn@split #1=\relax
1247     \expandafter\MT@kn@do
1248   \fi
1249 }
```

\MT@kn@split

```
1250 \def\MT@kn@split#1=#2=#3\relax{%
1251   \def\@tempa{#1}%
1252   \ifx\@tempa\@empty \else
1253     \MT@get@slot
1254     \ifnum\MT@char > \m@ne
1255       \MT@get@char@unit
1256       \MT@kn@split@val#2\relax
1257     \fi
1258   \fi
1259 }
```

\MT@kn@split@val Again, the unit may be measured in the space dimension; this time only \fontdimen2.

```
1260 \def\MT@kn@split@val#1,#2\relax{%
1261   \def\@tempb{#1}%
1262   \MT@ifempty\@tempb\relax{%
1263     \MT@get@space@unit\tw@
1264     \MT@scale@to@em
1265     \knbccode\MT@font\MT@char=\@tempcntb
1266     <debug>\MT@dinfo@n1{4}{;; knbc (\MT@char): \number\knbccode\MT@font\MT@char: [#1]}%
1267   }%
1268   \def\@tempb{#2}%
1269   \MT@ifempty\@tempb\relax{%
1270     \MT@get@space@unit\tw@
1271     \MT@scale@to@em
1272     \knaccode\MT@font\MT@char=\@tempcntb
1273     <debug>\MT@dinfo@n1{4}{;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1274   }%
1275   \MT@ifdefined@c@T\MT@kn@inh@name{%
1276     \MT@ifdefined@n@T{\MT@inh@\MT@kn@inh@name @\MT@char @}{%
1277       \expandafter\MT@map@tlist@c
1278       \csname MT@inh@\MT@kn@inh@name @\MT@char @\endcsname
1279       \MT@set@kn@heirs
1280     }%
1281   }%
1282 }
```

\MT@set@kn@heirs

```
1283 \def\MT@set@kn@heirs#1{%
1284   \knbccode\MT@font#1=\knbccode\MT@font\MT@char
1285   \knaccode\MT@font#1=\knaccode\MT@font\MT@char
1286   <*debug>
1287   \MT@dinfo@n1{2}{-- heir of \MT@char: #1}%
1288   \MT@dinfo@n1{4}{;; knbc (#1): \number\knbccode\MT@font\MT@char/%
1289                                     \number\knaccode\MT@font\MT@char}%
1290   </debug>
1291 }
```

\MT@set@all@kn

\MT@reset@kn@codes 1292 \def\MT@set@all@kn#1#2{%  
\MT@reset@kn@codes@

```

1293 <debug>\MT@info@n1{3}{-- knac/knbc: setting all to \number#1/\number#2}%
1294 \MT@do@font{%
1295 \knbccode\MT@font\@tempcnta=#1\relax
1296 \knaccode\MT@font\@tempcnta=#2\relax
1297 }%
1298 }
1299 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1300 \let\MT@reset@kn@codes\relax

\MT@preset@kn
\MT@preset@kn@ 1301 \def\MT@preset@kn{%
1302 \expandafter\expandafter\expandafter\MT@preset@kn@
1303 \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1304 }
1305 \def\MT@preset@kn@#1,#2\@nil{%
1306 \ifx\MT@kn@unit@\@empty
1307 \MT@warn@preset@twidth{kn}%
1308 \let\MT@preset@aux\MT@preset@aux@factor
1309 \else
1310 \let\MT@preset@aux\MT@preset@aux@space
1311 \fi

Letterspacing factor.
1312 \MT@ifstreq\MT@kn@context{letterspacing}{%
1313 \@tempcnta\MT@kn@factor\relax
1314 \MT@scale\@tempcnta \MT@letterspacing@ \@m
1315 \edef\MT@kn@factor@{\number\@tempcnta}%
1316 }\relax
1317 \MT@preset@aux{#1}\@tempa
1318 \MT@preset@aux{#2}\@tempb
1319 \MT@set@all@kn\@tempa\@tempb
1320 }

```

### 13.4.5 Letterspacing

`\lsstyle` Letterspacing is a special case of extra kerning. It will temporarily switch kerning on, activate the all font set and load the letterspacing context. The list must have the name `letterspacing`, so that the factor will be applied.

```

1321 \MT@requires@pdfTeX6{%
1322 \DeclareRobustCommand\lsstyle{%
1323 \ifMT@kerning

```

We have to add the current font to the active kerning font set, so that it is guaranteed to be reset. This will fail for font switches inside `\lsstyle`.

```

1324 \begingroup
1325 \escapechar\m@ne
1326 \expandafter\MT@exp@two@n\expandafter\MT@in@clist
1327 \csname\curr@fontshape/\f@size\expandafter\endcsname
1328 \csname MT@knlist@font@\MT@kn@setname\endcsname
1329 \ifMT@inlist@ \else
1330 \expandafter\MT@xadd
1331 \csname MT@knlist@font@\MT@kn@setname\endcsname
1332 {\MT@exp@string\csname\curr@fontshape/\f@size\endcsname,}%
1333 \fi
1334 \endgroup
1335 \fi
1336 \MT@kerningtrue
1337 \pdfappendkern\@ne

```

```

1338 \pdfprependkern\@ne
1339 \def\MT@kn@setname{all}%
1340 \MT@ifdefined@c@TF\MT@letterspacing@relax{%
1341   \let\MT@letterspacing@\MT@letterspacing
1342 }%
1343 \microtypecontext{kerning=letterspacing}%
1344 }
1345 }%
1346 \DeclareRobustCommand\lsstyle{%
1347   \MT@warning{Letterspacing only works with pdftex version 1.4x\MessageBreak
1348     or newer. You might want to use the 'soul' package\MessageBreak
1349     instead}%
1350   \global\let\lsstyle\relax
1351 }
1352 }

```

`\textls` This command may be used like the other text commands. The optional argument `\MT@letterspacing@` may be used to change the letterspacing factor.

```

1353 \DeclareRobustCommand\textls[2][ ]{%
1354   \MT@ifempty{#1}{%
1355     \let\MT@letterspacing@\@undefined
1356   }{%
1357     \KV@sp@def\MT@letterspacing@{#1 }%
1358   }%
1359   {\lsstyle #2}%
1360 }
1361 /beta

```

### 13.4.6 Disabling Ligatures

`\MT@noligatures` The possibility to disable ligatures is a new features of pdf<sub>T</sub><sub>E</sub>X 1.30.0.

```

1362 \MT@requires@pdftex5{%
1363   \def\MT@noligatures{%
1364     \csname ifMT@\MT@abbr@n1\endcsname
1365     \MT@dotrue
1366     \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1367       \MT@ifdefined@n@TF{MT@checklist@##1}%
1368       {\csname MT@checklist@##1\endcsname}%
1369       {\MT@checklist@{##1}}%
1370     {n1}%
1371   }%
1372   \else
1373     \MT@dofalse
1374   \fi
1375   \ifMT@do
1376     \pdfnoligatures\MT@font
1377     \MT@vinfo{... Disabling ligatures}%
1378   \fi
1379 }
1380 }%
1381 \let\MT@noligatures\relax
1382 }

```

### 13.4.7 Loading the Configuration

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

```

1383 \def\MT@load@list#1{%

```



```

1384 \edef\@tempa{#1}%
1385 \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa load}%
1386 \MT@ifstreq\@tempa\@tempb{%
1387   \MT@warning{\@nameuse{MT@abbr@\MT@feat} list '\@tempa' cannot load itself}%
1388 }{%
1389   \ifx\@tempb\relax \else
1390     \MT@ifdefined@n@TF{MT@MT@feat @c@\@tempb}{%
1391       \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list '\@tempb'}%
1392       \begingroup
1393         \MT@load@list{\@tempb}%
1394       \endgroup
1395       \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list '\@tempb'}%
1396       \MT@let@cn\@tempc{MT@MT@feat @c@\@tempb}%
1397       \expandafter\csname MT@MT@feat @do\expandafter\endcsname\@tempc,\relax,%
1398     }{%
1399       \MT@warning{\@nameuse{MT@abbr@\MT@feat} list '\@tempb' undefined.
1400         Cannot load\MessageBreak it from list '\@tempa'}%
1401     }%
1402   \fi
1403 }%
1404 }

```

\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.

```

1405 \let\MT@file@list\empty
1406 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

1407 \MT@in@clist{#1}\MT@file@list
1408 \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.

```

1409 \MT@begin@catcodes
1410 \let\MT@begin@catcodes\relax
1411 \let\MT@end@catcodes\relax
1412 \InputIfFileExists{mt-#1.cfg}{%
1413   \edef\MT@curr@file{mt-#1.cfg}%
1414   \MT@vinfo{... Loading configuration file \MT@curr@file}%
1415   \MT@xadd\MT@file@list{#1}%
1416 }{%
1417   \expandafter\MT@get@basefamily#1\relax\relax\relax
1418   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1419   \ifMT@inlist\else
1420     \InputIfFileExists{mt-\@tempa.cfg}{%
1421       \edef\MT@curr@file{mt-\@tempa.cfg}%
1422       \MT@vinfo{... Loading configuration file \MT@curr@file}%
1423       \MT@xadd\MT@file@list{\@tempa,#1}%
1424     }{%
1425       \MT@vinfo{... No configuration file mt-#1.cfg}%
1426       \MT@xadd\MT@file@list{#1}%
1427     }%
1428   \fi
1429 }%
1430 \endgroup
1431 \fi
1432 }

```

\MT@begin@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.

We leave ^ at catcode 7, so that stuff like `^^ff` remains possible.

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.

```

1433 \def\MT@begin@catcodes{%
1434   \begingroup
1435   \makeatletter
1436   \catcode'\^7%
1437   \catcode'\_9%
1438   \catcode'\^^I9%
1439   \catcode'\^^M9%
1440   \catcode'\\\z@
1441   \catcode'\{\@ne
1442   \catcode'\}\tw@
1443   \catcode'\#6%
1444   \catcode'\%14%
1445   \MT@map@tlist@n
1446   {\!\"\$&\'(\)\*+,\-\.\/\:\;\<=\>\?[\]\_\'|\|~}%
1447   \@makeother

```

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

```

1448   \let\KV@@sp@def\def
1449 }

```

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

```

1450 \let\MT@end@catcodes\endgroup

```

`\MT@get@basefamily` The family name might have a suffix for expert or old style number font set or for swash capitals (x, j or w). 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 ...).

```

1451 \def\MT@get@basefamily#1#2#3#4\relax{%
1452   \ifx#2\relax \def\@tempa{#1}\else
1453     \ifx#3\relax \def\@tempa{#1#2}\else
1454       \def\@tempa{#1#2#3}%
1455     \ifx\relax#4\relax \else
1456       \MT@ifstreq{#4}{\string x}\relax{%
1457         \MT@ifstreq{#4}{\string j}\relax{%
1458           \MT@ifstreq{#4}{\string w}\relax{%
1459             \def\@tempa{#1#2#3#4}}}\fi\fi\fi
1460 }

```

`\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@ 1461 \def\MT@get@listname#1{%
1462   (debug) \MT@info@n1{1}{trying to find \@nameuse{MT@abbr@#1} list for font \MT@font}%
1463   \let\MT@listname\@undefined
1464   \def\@tempb{#1}%
1465   \MT@map@tlist@c\MT@try@order\MT@get@listname@
1466 }
1467 \def\MT@get@listname@#1{%
1468   \expandafter\MT@next@listname#1%

```

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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```

1469 \ifx\MT@listname\undefined \else
1470 \expandafter\MT@tlist@break
1471 \fi
1472 }

```

\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.

```

1473 \def\MT@try@order{%
1474 {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
1475 {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
1476 }

```

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

```

1477 \def\MT@next@listname#1#2#3#4{%
1478 \edef\@tempa{\MT@encoding
1479 \ifnum#1=\@ne \MT@family\fi
1480 \ifnum#2=\@ne \MT@series\fi
1481 \ifnum#3=\@ne \MT@shape\fi
1482 \ifnum#4=\@ne *\fi
1483 \MT@context}%
1484 <debug>\MT@dinfo@n1{1}{trying \@tempa}%
1485 \MT@ifdefined@n@TF{\MT@\@tempb @\@tempa}{%
1486 \MT@next@listname@#4%
1487 }{%

```

Also try with an alias family.

```

1488 \ifnum#1=\@ne
1489 \ifx\MT@familyalias\@empty \else
1490 \edef\@tempa{\MT@encoding
1491 \MT@familyalias
1492 \ifnum#2=\@ne \MT@series\fi
1493 \ifnum#3=\@ne \MT@shape\fi
1494 \ifnum#4=\@ne *\fi
1495 \MT@context}%
1496 <debug>\MT@dinfo@n1{1}{(alias) \@tempa}%
1497 \MT@ifdefined@n@TF{\MT@\@tempb @\@tempa}{%
1498 \MT@next@listname@#4%
1499 }%
1500 \fi
1501 \fi
1502 }%
1503 }

```

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

```

1504 \def\MT@next@listname@#1{%

```

```

1505 \ifnum#1=\@ne
1506 \expandafter\MT@in@rlist\csname MT@\@tempb @\@tempa @sizes\endcsname
1507 \ifMT@inlist@
1508 \let\MT@listname\MT@size@name
1509 \fi
1510 \else
1511 \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
1512 \fi
1513 }

```

\MT@if@list@exists

```

\MT@context 1514 \def\MT@if@list@exists{%
1515 \MT@let@cn\MT@context{MT@\MT@feat @context}%
1516 \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
1517 \MT@get@listname{\MT@feat @c}%
1518 \MT@ifdefined@c@TF\MT@listname{%
1519 \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
1520 \ifMT@nonselected
1521 \MT@vinfo{... Applying non-selected expansion (list '\MT@ex@c@name')}}%
1522 \else
1523 \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list
1524 '\@nameuse{MT@\MT@feat @c@name}}}%
1525 \fi
1526 \@firstoftwo
1527 }{%

```

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

```
1528 \MT@let@nc{MT@\MT@feat @c@name}\@empty
```

Don't warn if selected=false.

```

1529 \ifMT@nonselected
1530 \MT@vinfo{... Applying non-selected expansion}%
1531 \else
1532 \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
1533 for font\MessageBreak'\MT@exp@string\MT@font'%
1534 \ifx\MT@context\@empty\else\space(context: '\MT@context')}\fi.
1535 Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
1536 \fi
1537 \@secondoftwo
1538 }%
1539 }

```

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

```

\MT@context 1540 \def\MT@get@inh@list{%
1541 \let\MT@context\@empty
1542 \MT@get@listname{\MT@feat @inh}%
1543 \MT@ifdefined@c@TF\MT@listname{%
1544 \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
1545 <debug>
1546 \MT@dinfo@n1{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
1547 '\@nameuse{MT@\MT@feat @inh@name}}}%
1548 </debug>
1549 \MT@let@cn\@tempc{MT@\MT@feat @inh\csname MT@\MT@feat @inh@name\endcsname}%

```

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

```

1550 \ifx\@tempc\@empty \else
1551 <debug>\MT@dinfo@n1{1}{parsing inheritance list ...}%
1552 \MT@let@cn\MT@inh@name{MT@\MT@feat @inh@name}%

```

```

1553     \def\MT@curr@list@name{inheritance list\MessageBreak'\MT@inh@name'}%
1554     \expandafter\MT@inh@do\@tempc,\relax,%
1555     \global\MT@let@nc{\MT@MT@feat @inh@csname MT@\MT@feat @inh@name\endcsname}\@empty
1556     \fi
1557   }{%
1558     \MT@let@nc{\MT@MT@feat @inh@name}\@undefined
1559   }%
1560 }

```

### 13.4.8 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.

The character is in \@tempa, we want its slot number in \MT@char.

```

1561 \def\MT@get@slot{%
1562   \escapechar'\
1563   \let\Mt@char\m@ne
1564   \MT@noresttrue

```

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

```

1565   \MT@toks=\expandafter\expandafter\expandafter{\expandafter\string\@tempa}%
1566   \edef\MT@char{\expandafter\meaning\@tempa}%

```

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

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

```

1567     \expandafter\MT@is@letter\@tempa\relax\relax
1568     \ifnum\Mt@char < \z@

```

- It might be an active character, i. e., an 8-bit character defined by inputenc.

```

1569     \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.

```

1570     \MT@ifdefined@n@TF{\MT@encoding\Mt@detokenize\@tempa}%
1571     \MT@is@symbol
1572   {%

```

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

```

1573     \expandafter\MT@is@composite\@tempa\relax\relax
1574   }%
1575   \ifnum\Mt@char < \z@

```

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

```

1576     \MT@exp@two@c\Mt@is@char\Mt@char\Mt@charstring\relax\relax\relax

```

```

1577     \fi
1578     \fi

1579     \let\Mt@char\Mt@char
1580     \ifnum\Mt@char < \z@
1581         \MT@warn@unknown
1582     \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.

```

1583     \ifMT@norest \else
1584         \MT@warn@unknown@i
1585         \let\Mt@char@m@ne
1586     \fi
1587     \fi
1588     \escapechar@m@ne
1589 }

```

**\ifMT@norest** Switch and test whether all of the string has been used up.

```

\MT@testrest 1590 \newif\ifMT@norest
1591 \def\MT@testrest#1#2{%
1592     \MT@ifstreq{#1}{#2}\relax\MT@norestfalse
1593 }

```

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

```

1594 \def\MT@is@letter#1#2\relax{%
1595     \ifcat a\noexpand#1\relax
1596         \edef\Mt@char{\number'#1}%
1597         \ifx\#2\%
1598             (debug) \MT@dinfo@n1{3}{> '\the\MT@toks' is a letter (\Mt@char)}%
1599         \else
1600             \MT@norestfalse
1601         \fi
1602     \else
1603         \ifcat 1\noexpand#1\relax
1604             \edef\Mt@char{\number'#1}%
1605             (debug) \MT@dinfo@n1{3}{> '\the\MT@toks' is a character (\Mt@char)}%
1606             \ifx\#2\%
1607                 \ifnum\Mt@char>127 \MT@warn@ascii \fi
1608             \else
1609                 \MT@norestfalse
1610                 \expandafter\Mt@is@number#1#2\relax\relax
1611             \fi
1612         \fi
1613     \fi
1614 }

```

**\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.

```

1615 \def\MT@is@number#1#2#3\relax{%
1616     \ifx\relax#3\relax \else
1617         \ifx\relax#2\relax \else
1618             \MT@noresttrue
1619             \if#1"\relax
1620                 \def\x{\uppercase{\edef\Mt@char{\number#1#2#3}}}\x

```

```

1621 <debug> \MT@info@n1{3}{> ... a hexadecimal number: \Mt@char}%
1622     \else
1623         \if#1'\relax
1624             \def\Mt@char{\number#1#2#3}%
1625 <debug> \MT@info@n1{3}{> ... an octal number: \Mt@char}%
1626     \else
1627         \MT@ifnumber{#1#2#3}{%
1628             \def\Mt@char{\number#1#2#3}%
1629 <debug> \MT@info@n1{3}{> ... a decimal number: \Mt@char}%
1630         }\MT@norestfalse
1631     \fi
1632 \fi
1633 \ifnum\Mt@char > \ecclv
1634     \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
1635     \let\Mt@char\m@ne
1636 \fi
1637 \fi
1638 \fi
1639 }

```

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

Previous solution, slightly more robust:

```

\def\MT@is@active#1#2\@nil{%
  \ifx\#2\%
    \ifnum\catcode'#1 = \active
      \toks@=\expandafter\expandafter\expandafter{\@tempa}%
      \expandafter\MT@active@inpenc\the\toks@\relax\relax
      \edef\@tempa{\the\toks@}%
      \edef\x{\MT@toks={\the\MT@toks\space(= \the\toks@)}}\x
    \fi
  \fi
}
\def\MT@active@inpenc#1#2#3\relax{%
  \ifx#1\IeC
    \def\IeC##1{\toks@={##1}}%
    \the\toks@
    \expandafter\MT@active@inpenc\the\toks@\relax\relax
  \fi
  \ifx#1\@tabacckludge
    \def\@tabacckludge##1##2{%
      \toks@=\expandafter{\csname\string##1\endcsname#3}}%
    \the\toks@
  \fi
  \ifx#1\@inpenc@undefined@
    \def\@inpenc@undefined@##1{%
      \edef\x{\toks@={%
        undefined^^J(\MT@MT)\@spaces\@spaces\@spaces\@spaces
        in input encoding '"#1'}}\x}%
    \the\toks@
  \fi
}

```

Unfortunately, the `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 currently not supported.

```
1640 \def\MT@is@active#1#2\@nil{%
1641   \ifx\#2\%
1642     \ifnum\catcode'#1 = \active
1643       \begingroup
1644         \set@display@protect
1645         \def\IeC#1{##1}%
```

The character is undefined in the encoding.

```
1646     \def\@inpenc@undefined@##1{undefined^^J%
1647       (\MT@MT)\@spaces\@spaces\@spaces\@spaces
1648       in input encoding '##1'}%
1649     \edef\x{%
1650       \def\noexpand\@tempa{\@tempa}%
```

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

```
1651     \MT@toks={\the\MT@toks\space (= \@tempa)}%
1652   }%
1653   \expandafter
1654   \endgroup
1655   \x
1656   \fi
1657   \fi
1658 }
```

`\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.

```
1659 \def\MT@is@symbol{%
1660   \edef\@tempa{\expandafter
1661     \csname\expandafter
1662       \MT@encoding\expandafter
1663       \string\@tempa
1664       \endcsname}%
1665   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1666   \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
1667   \ifnum\Mt@char < \z@
```

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

```
1668     \expandafter\MT@is@letter\@tempa\relax\relax
1669     \fi
1670 }
```

`\MT@is@char` Here we define a helper macro that inspect the `\meaning` of its argument.

```
\MT@charstring 1671 <debug>\def\Mt@info{\MT@dinfo@n1{3}}
1672 \begingroup
1673   \catcode'\/=0
1674   /MT@map@tlist@n{/\/C/H/A/R}/@makeother
1675   /lowercase{%
1676     /def/x{%
1677       /def/MT@charstring{\CHAR"%
1678       /def/MT@is@char##1\CHAR"##2##3##4/relax{%
1679         /ifx/relax##1/relax
```



```

1680      /if##3\relax
1681      /edef\Mt@char{/number"##2}%
1682      /MT@testrest/MT@charstring{##3##4}%
1683      /else
1684      /edef\Mt@char{/number"##2##3}%
1685      /MT@testrest/MT@charstring{##4}%
1686      /fi
1687 (debug) /Mt@info{> '/the/MT@toks' is a \char (/Mt@char)}%
1688      /fi
1689      }%
1690      }%
1691      }
1692 /expandafter/endgroup/x

```

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

```

1693 \def\MT@is@composite#1#2\relax{%
1694   \ifx\#2\\\else

```

Again, we construct a control sequence, this time of the form: `\\<encoding>  
\\<accent>-<character>`, e. g. `\\T1\"-a`, which expands to a letter if it has been defined by `\DeclareTextComposite`. This should be robust, finally.

```

1695   \edef\@tempa{\expandafter
1696               \csname\expandafter
1697                   \string\csname\MT@encoding\endcsname
1698                   \string#1-%
1699                   \string#2%
1700                   \endcsname}%
1701   \expandafter\MT@is@letter\@tempa\relax\relax
1702   \fi
1703 }

```

`\MT@detokenize` Translate a macro into a token list. We must be cautious not to stumble over accented characters consisting of two commands, like `'\i` or `\U\CYRI`. With e- $\TeX$ , we can use `\detokenize` (and `\expandafter\string` to get rid of the trailing space). The non-e- $\TeX$  version requires some more fiddling.

```

1704 \MT@requires@etex{%
1705   \def\MT@detokenize#1{\detokenize
1706       \expandafter\expandafter\expandafter{\expandafter\string#1}}
1707 }{%
1708   \def\MT@detokenize#1{\MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty}
1709 }

```

Some warning messages, for performance reasons separated here.

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

```

1710 \let\MT@curr@list@name\@empty

```

`\Mt@warn@ascii` For characters with character code > 127, we issue a warning (inputenc probably hasn't been loaded), since correspondence with the slot numbers would be purely coincidental.

```

1711 \def\Mt@warn@ascii{%
1712   \MT@warning@nl{Character '\the\MT@toks' (= \Mt@char)
1713       is outside of ASCII range.\MessageBreak
1714       You must load the 'inputenc' package before using\MessageBreak
1715       8-bit characters in \MT@curr@list@name}%
1716 }

```

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

```

1717 \def\MT@warn@number@too@large#1{%
1718   \MT@warning@nl{%
1719     Number #1 in encoding '\MT@encoding' too large!\MessageBreak
1720     Ignoring it in \MT@curr@list@name}%
1721 }

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

1722 \def\MT@warn@unknown@i{%
1723   \MT@warning@nl{%
1724     Unknown slot number of character '\the\MT@toks' in\MessageBreak
1725     font encoding '\MT@encoding'. Make sure it's a single\MessageBreak
1726     character (or a number) in \MT@curr@list@name}%
1727 }

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

1728 \def\MT@warn@unknown{%
1729   \MT@warning@nl{%
1730     Unknown slot number of character '\the\MT@toks' in\MessageBreak
1731     font encoding '\MT@encoding' in \MT@curr@list@name}%
1732 }

```

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

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

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

For my reference:

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

Up to version 1.3 of this package, we were using `\define@newfont` as the hook, which is only called for *new* fonts, and therefore seemed the natural choice. However, this meant that we had to take special care to catch all fonts: We additionally had to set up the default font, the error font (if it wasn't the default font), we had to check for some packages that might have been loaded before microtype and were loading fonts, e.g. `jurabib`, `ledmac`, `pifont` (loaded by `hyperref`), `tifa`,

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 favorite class, the memoir class, 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.

```
1733 \let\MT@font@list\empty
```

\MT@font 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.

```
1734 \let\MT@font\empty
1735 \MT@addto@setup{%
1736   \g@addto@macro\do@subst@correction{%
1737     \xdef\MT@font{\csname \curr@fontshape/\f@size\endcsname}%
1738   }
```

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

```
1739 \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
1740 \ifx\pickup@font\MT@orig@pickupfont \else
1741   \MT@warning@nl{%
1742     Command \string\pickup@font\space is not defined as expected.\MessageBreak
1743     Double-check whether micro-typography is indeed.\MessageBreak
1744     applied to the document.\MessageBreak (Hint: Turn on 'verbose' mode)%
1745   }
1746 \fi
```

Then we append our stuff.

```
1747 \g@addto@macro\pickup@font{%
1748   \begingroup
1749     \escapechar\m@ne
```

If \MT@font@name 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 one-level substitutions, like bx to b, but it will still fail if the substituting font is itself substituted.

```
1750   \ifx\MT@font\empty
1751     \let\MT@font\font@name
1752   \else
1753     \ifx\MT@font\font@name \else
1754       \MT@register@subst@font
1755     \fi
1756   \fi
1757   \MT@check@font
1758   \ifMT@inlist@ \else
1759     \MT@setupfont
1760     \MT@register@font
1761   \fi
1762 \endgroup
1763 \global\let\MT@font\empty
1764 }
```

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

```
1765 \let\MT@pickupfont\pickupfont
```

`\MT@orig@add@accent` Inside `\add@accent`, we have to disable microtype's setup, since the grouping in the patched `\pickupfont` 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. At first, I was going to change `\hmode@bgroup` only, but that is also used in the commands defined by `\DeclareTextFontCommand`, i. e., `\textit` etc.

```
1766 \let\MT@orig@add@accent\add@accent
1767 \def\add@accent#1#2{%
1768   \let\pickupfont\MT@orig@pickupfont
1769   \MT@orig@add@accent{#1}{#2}%
1770   \let\pickupfont\MT@pickupfont
1771 }
1772 }
```

Consequently, we are the last one to change these commands, therefore there is no need to check whether our definition has survived.<sup>15</sup>

`\MT@check@font` Check whether we've already seen the current font.

```
1773 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

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

```
1774 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
```

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

```
1775 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}
```

### 13.4.10 Context-sensitive Setup

Here are the variants for context-sensitive setup.

`\MT@active@features` The activated features are stored in this command.

```
1776 \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.

```
1777 \def\MT@check@font@cx{%
1778   \@tempswatrue
1779   \MT@map@clist@cx\MT@active@features{%
1780     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
1781     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1782     \ifMT@inlist@
1783       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
1784     \else
1785       \@tempswafalse
1786     \fi
1787   }%
1788   \if@tempswa \MT@inlist@true \else \MT@inlist@false \fi
1789 }
```

<sup>15</sup> I still don't know whether microtype works together with CJK

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

```
1790 \def\MT@register@subst@font@cx{%
1791   \MT@map@clist@c\MT@active@features{%
1792     \expandafter\MT@xadd
1793     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1794     {\font@name,}%
1795   }%
1796 }
```

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

```
1797 \def\MT@register@font@cx{%
1798   \MT@map@clist@c\MT@active@features{%
1799     \expandafter\ifx\csname MT@\@nameuse{MT@abbr@##1}\endcsname\relax\else
1800     \expandafter\MT@xadd
1801     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1802     {\MT@font,}%
1803     \def\@tempa{##1}%
1804     \expandafter\MT@map@tlist@c
1805     \csname MT@##1@doc@contexts\endcsname
1806     \MT@rem@from@lists
1807   \fi
1808 }%
1809 }
```

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

```
1810 \def\MT@rem@from@lists#1{%
1811   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
1812     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist
1813     \expandafter\MT@font\csname MT@\@tempa @#1font@list\endcsname
1814   }%
1815 }
```

`\microtypecontext` The user may change the context, so that different setups are possible. This is especially useful for multi-lingual documents.

Inside the preamble, it shouldn't actually do anything but remember it for later.

```
1816 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
1817 \MT@addto@setup{%
1818   \def\microtypecontext#1{%
1819     \MT@setup@contexts
1820     \setkeys{MTC}{#1}%
1821     \selectfont
1822     \aftergroup\MT@reset@context
1823   }%
1824 }
```

`\MT@reset@context` We have to reset the font at the end of the group.

```
1825 \def\MT@reset@context{%
1826   \MT@vinfo{Resetting contexts on line \the\inputlineno
1827   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
1828   <debug&beta> / \MT@kn@context/\MT@sp@context
1829   }%
1830   \selectfont
1831 }
```

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

```
1832 \def\MT@setup@contexts{%
```

```

1833 \MT@map@clist@{MT@active@features}%
1834 \global\MT@let@enc{MT@#1@font@list}\MT@font@list
1835 }%
1836 \global\let\MT@check@font\MT@check@font@cx
1837 \global\let\MT@register@font\MT@register@font@cx
1838 \global\let\MT@register@subst@font\MT@register@subst@font@cx
1839 \global\let\MT@setup@contexts\relax
1840 }

```

\MT@define@context

```

1841 \def\MT@define@context#1{%
1842 \define@key{MTC}{#1}[] {%
1843 \KV@sp@def\@tempb{#1}%
1844 \edef\@tempb{\@nameuse{MT@rbba@\@tempb}}%
1845 \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
1846 \ifMT@inlist@

```

Using an empty context is only asking for trouble, therefore we use the ‘@’ instead.

```

1847 \MT@ifempty{##1}{\def\MT@val{0}}{\KV@sp@def\MT@val{##1}}%
1848 \expandafter\ifx\csname MT@\@tempb @context\endcsname\MT@val \else
1849 \MT@vinfo{--- Changing #1 context to '\MT@val'}%

```

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

```

1850 \global\MT@let@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes}%

```

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

```

1851 \expandafter\MT@exp@one@n\expandafter\MT@in@tlist
1852 \expandafter\MT@val\csname MT@\@tempb @doc@contexts\endcsname
1853 \ifMT@inlist@ \else
1854 \expandafter\MT@xadd\csname MT@\@tempb @doc@contexts\endcsname{\MT@val}%
1855 (debug) \MT@dinfo{2}{>>> added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
1856 \fi
1857 \MT@edef{n{MT@\@tempb @context}}{\MT@val}%
1858 \fi
1859 \fi
1860 }%
1861 }

1862 \MT@define@context{protrusion}
1863 \MT@define@context{expansion}
1864 (beta)
1865 \MT@define@context{spacing}
1866 \MT@define@context{kerning}
1867 (beta)

```

\MT@pr@context Initialize the contexts.

```

\MT@ex@context 1868 \def\MT@pr@context{@}
\MT@sp@context 1869 \def\MT@ex@context{@}
\MT@kn@context 1870 \def\MT@n@context{@}
1871 \global\def\MT@pr@doc@contexts{{@}}
\MT@pr@doc@contexts 1872 \global\def\MT@ex@doc@contexts{{@}}
\MT@ex@doc@contexts 1873 \global\def\MT@n@doc@contexts{{@}}
1874 (beta)
\MT@sp@doc@contexts 1875 \def\MT@sp@context{@}
\MT@kn@doc@contexts 1876 \def\MT@kn@context{@}
1877 \global\def\MT@sp@doc@contexts{{@}}
\MT@extra@context 1878 \global\def\MT@kn@doc@contexts{{@}}
1879 (beta)
1880 \let\MT@extra@context\empty

```

## 13.5 Configuration

### 13.5.1 Font Sets

`\DeclareMicrotypeSet` Calling this macro will create a comma list for every font characteristic of the form: `\MT<feature>list@<characteristic>@<set name>`. If the optional argument is empty, `\DeclareMicrotypeSet*` lists for both expansion and protrusion will be created.

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

```
1881 \def\DeclareMicrotypeSet{%
1882   \ifstar
1883     {\ifnextchar[\MT@DeclareSetAndUseIt
1884       {\MT@DeclareSetAndUseIt[]}}%
1885     {\ifnextchar[\MT@DeclareSet
1886       {\MT@DeclareSet[]}}%
1887 }
```

```
\MT@DeclareSet
\MT@DeclareSetAndUseIt 1888 \def\MT@DeclareSet[#1]{%
1889   \MT@DeclareSet@{#1}%
1890 }
1891 \def\MT@DeclareSetAndUseIt[#1]#2#3{%
1892   \MT@DeclareSet@{#1}{#2}{#3}%
1893   \UseMicrotypeSet[#1]{#2}%
1894 }
```

```
\MT@DeclareSet@
1895 \def\MT@DeclareSet@#1#2#3{%
1896   \KV@sp@def\@tempa{#1}%
1897   \MT@ifempty\@tempa{%
1898     \MT@declare@sets{pr}{#2}{#3}%
1899     \MT@declare@sets{ex}{#2}{#3}%
1900     <beta>
1901     \MT@declare@sets{sp}{#2}{#3}%
1902     \MT@declare@sets{kn}{#2}{#3}%
1903     /beta>
1904   }%
1905   \MT@map@c@list@c\@tempa{%
1906     \KV@sp@def\@tempa{##1}%
1907     \MT@ifempty\@tempa\relax{%
1908       \MT@is@feature{set declaration '#2'}{%
1909         \MT@exp@one@n\MT@declare@sets
1910         {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
1911       }%
1912     }%
1913   }}%
1914 }%
1915 }
```

`\MT@is@feature` Never trust user input: We check whether a feature exists to prevent a rather  
`\MT@allowed@features` confusing ‘Missing `\endcsname` inserted’ error message.

```
1916 \def\MT@is@feature#1{%
1917   \MT@ifdefined@n@TF{MT@rbba@\@tempa}\@firstofone{%
1918     \MT@error{'\@tempa' is not a defined feature.\MessageBreak
1919       Ignoring #1}{Defined features are: '\MT@allowed@features'.}%
1920     \@gobble}%
1921 }
```

```

1921 }
1922 \def\MT@allowed@features{protrusion,expansion%
1923 <beta>,spacing,kerning%
1924 }

```

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

```

1925 \let\MT@curr@set@name\empty

```

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

```

1926 \def\MT@declare@sets#1#2#3{%
1927   \KV@sp@def\MT@curr@set@name{#2}%
1928   \MT@ifdefined@nT{MT@#1@set@{\MT@curr@set@name}}{%
1929     \MT@warning{Redefining set '\MT@curr@set@name'}%
1930   }%
1931   \global\MT@let@nc{MT@#1@set@{\MT@curr@set@name}}\empty
1932 <debug>\MT@info{1}{declaring \nameuse{MT@abbr@#1} set '\MT@curr@set@name'}%
1933   \setkeys{MT@#1@set}{#3}%
1934 }

```

\MT@define@set@keys Define the keyval keys for expansion and protrusion sets.

```

1935 \def\MT@define@set@keys#1{%
1936   \MT@define@set@key@{encoding}{#1}%
1937   \MT@define@set@key@{family}{#1}%
1938   \MT@define@set@key@{series}{#1}%
1939   \MT@define@set@key@{shape}{#1}%
1940   \MT@define@set@key@size{#1}%
1941   \MT@define@set@key@font{#1}%
1942 }

```

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

```

1943 \def\MT@define@set@key@#1#2{%
1944   \define@key{MT@#2@set}{#1}[]{}%
1945   \global\MT@let@nc{MT@#2list@#1@{\MT@curr@set@name}}\empty
1946   \MT@map@clist@n{##1}{%
1947     \KV@sp@def\MT@val{####1}%
1948     \MT@get@highlevel{#1}%

```

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

```

1949     \MT@exp@two@n@g@addto@macro
1950     {\csname MT@#2list@#1@{\MT@curr@set@name}\expandafter\endcsname}%
1951     {\MT@val,}%
1952   }%

```

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

```

1953   \edef\x{%
1954     \expandafter\noexpand\expandafter\MT@fix@font@spec
1955     \expandafter\noexpand\csname MT@#2list@#1@{\MT@curr@set@name}\endcsname
1956   }%
1957   \MT@exp@one@n\MT@addto@setup\x
1958 <debug>\MT@info@n1{1}{-- #1: \nameuse{MT@#2list@#1@{\MT@curr@set@name}}}%
1959   }%
1960 }

```

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

```

1961 \def\MT@get@highlevel#1{%
1962   \expandafter\MT@test@ast\MT@val*\@nil{%

```

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



```

1963 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
1964 \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.

```

1965 }%
1966 }

```

**\MT@test@ast** Test whether last character is an asterisk.

```

1967 \def\MT@test@ast#1*#2\@nil{%
1968 \def\@tempa{#1}%
1969 \MT@ifempty{#2}%
1970 \@gobble
1971 \@firstofone
1972 }

```

**\MT@fix@font@spec** Fully expand the font specification and fix catcodes.

```

1973 \def\MT@fix@font@spec#1{%
1974 \xdef#1{#1}%
1975 \global\MT@make@string#1%
1976 }

```

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

```

1977 \def\MT@define@set@key@size#1{%
1978 \define@key{MT@#1@set}{size}[]{%
1979 \MT@map@clist@n{##1}{%
1980 \KV@sp@def\MT@val{###1}%
1981 \expandafter\MT@get@range\MT@val--\@nil
1982 \ifx\MT@val\relax \else
1983 \expandafter\MT@xadd
1984 \csname MT@#1list@size@\MT@curr@set@name\endcsname
1985 {{{\MT@lower}{\MT@upper}\relax}}}%
1986 \fi
1987 }%
1988 debug \MT@dinfo@n{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
1989 }%
1990 }

```

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 Minion-Pro 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 triples of  $\{\langle lower\ bound \rangle\}\{\langle upper\ bound \rangle\}\{\langle list\ name \rangle\}$ .  
**\MT@upper** For simple sizes, the upper boundary is -1.

```

\MT@lower 1991 \def\MT@get@range#1-#2-#3\@nil{%
1992 \MT@ifempty{#1}{%
1993 \MT@ifempty{#2}{%
1994 \let\MT@val\relax
1995 }{%
1996 \def\MT@lower{0}%
1997 \def\MT@val{#2}%
1998 \MT@get@size
1999 \edef\MT@upper{\MT@val}%
2000 }%
2001 }{%
2002 \def\MT@val{#1}%

```

```

2003 \MT@get@size
2004 \ifx\MT@val\relax \else
2005 \edef\MT@lower{\MT@val}%
2006 \MT@ifempty{#2}{%
2007 \MT@ifempty{#3}{%
2008 \def\MT@upper{-1}%
2009 }{%
2048 is TEX's maximum font size.
2010 \def\MT@upper{2048}%
2011 }%
2012 }{%
2013 \def\MT@val{#2}%
2014 \MT@get@size
2015 \ifx\MT@val\relax \else
2016 \MT@ifgt\MT@lower\MT@val{%
2017 \MT@warning{%
2018 Invalid size range (\MT@lower\space > \MT@val) in font set
2019 '\MT@curr@set@name'.\MessageBreak Swapping sizes}%
2020 \edef\MT@upper{\MT@lower}%
2021 \edef\MT@lower{\MT@val}%
2022 }{%
2023 \edef\MT@upper{\MT@val}%
2024 }%
2025 \MT@ifeq\MT@lower\MT@upper{%
2026 \def\MT@upper{-1}%
2027 }\relax
2028 \fi
2029 }%
2030 \fi
2031 }%
2032 }

```

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

```

2033 \def\MT@get@size{%

```

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

```

2034 \if*\MT@val\relax
2035 \def\@tempa{\normalsize}%
2036 \else
2037 \MT@let@cn\@tempa{\MT@val}%
2038 \fi
2039 \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).

```

2040 \begingroup
2041 \def\set@fontsize##1##2##3##4\@nil{\gdef\MT@val{##2}}%
2042 \@tempa\@nil
2043 \endgroup
2044 \fi

```

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

```

2045 \MT@ifdimen\MT@val{%

```

```

2046 \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2047 \edef\MT@val{\strip@pt\@tempdima}%
2048 }{%
2049 \MT@warning{Could not parse font size '\MT@val'\MessageBreak
2050 in font set '\MT@curr@set@name'}%
2051 \let\MT@val\relax
2052 }%
2053 }

```

\MT@define@set@key@font

```

2054 \def\MT@define@set@key@font#1{%
2055 \define@key{MT@#1@set}{font}[]{%
2056 \global\MT@let@nc{MT@#1list@font@\MT@curr@set@name}\@empty
2057 \MT@map@clist@n{##1}{%
2058 \KV@esp@def\MT@val{###1}%
2059 \expandafter\MT@get@font\MT@val////\@nil
2060 \MT@exp@two@n@g@addto@macro
2061 {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
2062 {\MT@val,}%
2063 }%
2064 \edef\x{%
2065 \expandafter\noexpand\expandafter\MT@fix@font@spec
2066 \expandafter\noexpand\csname MT@#1list@font@\MT@curr@set@name\endcsname
2067 }%
2068 \MT@exp@one@n\MT@addto@setup\x
2069 <debug>\MT@info@n1{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
2070 }%
2071 }

```

\MT@get@font Translate any asterisks.

```

2072 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2073 \MT@ifempty{#1#2#3#4#5}\relax{%
2074 \let\@tempb\@empty
2075 \def\MT@temp{#1/#2/#3/#4/#5}%
2076 \MT@get@axis{encoding}{#1}%
2077 \MT@get@axis{family}{#2}%
2078 \MT@get@axis{series}{#3}%
2079 \MT@get@axis{shape}{#4}%
2080 \MT@ifempty{#5}{%
2081 \MT@warn@axis@empty{size}{\string\normal size}%
2082 \def\MT@val{*}%
2083 }{%
2084 \def\MT@val{#5}%
2085 }%
2086 \MT@get@size
2087 \ifx\MT@val\relax\def\MT@val{0}\fi
2088 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2089 \let\MT@val\@tempb
2090 }%
2091 }

```

\MT@get@axis

```

2092 \def\MT@get@axis#1#2{%
2093 \def\MT@val{#2}%
2094 \MT@get@highlevel{#1}%
2095 \MT@ifempty\MT@val{%
2096 \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2097 \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2098 }\relax

```

```

2099 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2100 }

```

\MT@warn@axis@empty

```

2101 \def\MT@warn@axis@empty#1#2{%
2102   \MT@warning{#1 axis is empty in font specification\MessageBreak
2103     '\MT@temp'. Using '#2' instead}%
2104 }

```

We have finally assembled all pieces to define \DeclareMicrotypeSet's keys.

```

2105 \MT@define@set@keys{pr}
2106 \MT@define@set@keys{ex}
2107 <math>\langle\textit{beta}\rangle</math>
2108 \MT@define@set@keys{sp}
2109 \MT@define@set@keys{kn}
2110 <math>\langle\textit{beta}\rangle</math>

```

It is also used for \DisableLigatures.

```

2111 \MT@define@set@keys{nl}

```

\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.

```

2112 \renewcommand*\UseMicrotypeSet[2][]{%
2113   \KV@sp@def\@tempa{#1}%
2114   \MT@ifempty\@tempa{%
2115     \MT@use@set{pr}{#2}%
2116     \MT@use@set{ex}{#2}%
2117     <math>\langle\textit{beta}\rangle</math>
2118     \MT@use@set{sp}{#2}%
2119     \MT@use@set{kn}{#2}%
2120     <math>\langle\textit{beta}\rangle</math>
2121   }{%
2122     \MT@map@clistc\@tempa{%
2123       \KV@sp@def\@tempa{##1}%
2124       \MT@ifempty\@tempa\relax{%
2125         \MT@is@feature{activation of set '#2'}{%
2126           \MT@exp@one@n\MT@use@set
2127             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2128         }%
2129       }%
2130     }%
2131   }%
2132 }

```

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

```

\MT@ex@setname 2133 \def\MT@use@set#1#2{%
\MT@use@set 2134   \KV@sp@def\@tempa{#2}%
2135   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
2136     \global\MT@edef@n{MT@#1@setname}{\@tempa}%
2137     \MT@info{Using \@nameuse{MT@abbr@#1} set '\@tempa'}%
2138   }{%
2139     \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2140       \global\MT@edef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2141     }%
2142     \MT@warning{%
2143       The \@nameuse{MT@abbr@#1} set '\@tempa' is undeclared.\MessageBreak
2144       Using set '\@nameuse{MT@#1@setname}' instead}%
2145   }%
2146 }

```

`\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.

```

2147 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2148   \KV@sp@def\@tempa{#1}%
2149   \MT@ifempty\@tempa{%
2150     \MT@set@default@set{pr}{#2}%
2151     \MT@set@default@set{ex}{#2}%
2152   }(beta)
2153   \MT@set@default@set{sp}{#2}%
2154   \MT@set@default@set{kn}{#2}%
2155 }(beta)
2156 }{%
2157   \MT@map@clist@c\@tempa{%
2158     \KV@sp@def\@tempa{#1}%
2159     \MT@ifempty\@tempa\relax{%
2160       \MT@is@feature{declaration of default set '#2'}{%
2161         \MT@exp@one@n\MT@set@default@set
2162         {\csname MT@rbba@\@tempa\endcsname}{#2}%
2163       }%
2164     }%
2165   }%
2166 }%
2167 }

\MT@default@pr@set
\MT@default@ex@set 2168 \def\MT@set@default@set#1#2{%
\MT@default@kn@set 2169   \KV@sp@def\@tempa{#2}%
2170   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@default@sp@set 2171   }(debug)\MT@info{1}{declaring default \@nameuse{MT@abbr@#1} set '\@tempa'}%
\MT@set@default@set 2172   \global\MT@edef@n{MT@default@#1@set}{\@tempa}%
2173   }{%
2174     \MT@warning{%
2175       The \@nameuse{MT@abbr@#1} set '\@tempa' is not declared.\MessageBreak
2176       Cannot make it the default set. Using set\MessageBreak 'all' instead}%
2177     \global\MT@edef@n{MT@default@#1@set}{all}%
2178   }%
2179 }
```

`\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.

```

2180 \renewcommand*\DeclareMicrotypeAlias[2][]{%
2181   \KV@sp@def\@tempa{#1}%
2182   \KV@sp@def\@tempb{#2}%
2183   \MT@make@string\@tempb
2184   \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2185     \MT@warning{Alias font family '\@tempb' will override
2186       alias \@nameuse{MT@\@tempa @alias}.\MessageBreak
2187       for font family '\@tempa'}}%
2188   \global\MT@edef@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.

```

2189   \MT@ifdefined@c@T\MT@family{%
2190   }(debug)\MT@info{1}{Activating alias font '\@tempb' for '\MT@family'}%
2191   \global\let\MT@familyalias\@tempb
2192   }%
2193 }
```

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

```

2194 \def\LoadMicrotypeFile#1{%
2195   \KV@sp@def\@tempa{#1}%
2196   \MT@make@string\@tempa
2197   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2198   \ifMT@inlist@
2199     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2200   \else
2201     \MT@xadd\MT@file@list{\@tempa,}%
2202     \MT@begin@catcodes
2203     \InputIfFileExists{mt-\@tempa.cfg}{%
2204       \edef\MT@curr@file{mt-\@tempa.cfg}%
2205       \MT@vinfo{... Loading configuration file \MT@curr@file}%
2206     }{%
2207       \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2208         does not exist}%
2209     }%
2210     \MT@end@catcodes
2211   \fi
2212 }
```

`\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'.

```

2213 \MT@requires@pdftex5{%
2214   \renewcommand*\DisableLigatures[1]{%
2215     \edef\MT@active@features{\MT@active@features,nl}%
2216     \MT@noligaturestrue
2217     \MT@declare@sets{nl}{no ligatures}{#1}%
2218     \gdef\MT@nl@setname{no ligatures}%
2219   }
2220 }
```

If pdfTeX is too old, we issue a warning and neutralize the command.

```

2221 \renewcommand*\DisableLigatures[1]{%
2222   \MT@warning{Disabling ligatures of a font is only possible\MessageBreak
2223     with pdftex version 1.30 or later.\MessageBreak
2224     Ignoring \string\DisableLigatures}%
2225   \global\let\DisableLigatures@gobble
2226 }
2227 }
```

### 13.5.2 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.

```

2228 <beta>
2229 \def\DeclareMicrotypeBabelHook#1#2{%
2230   \MT@map@clist@n{#1}{%
2231     \KV@sp@def\@tempa{##1}%
2232     \global\MT@def@n\MT@babel@\@tempa}{#2}%
2233   }%
2234 }
2235 </beta>
```

### 13.5.3 Fine Tuning

The macros `\SetExpansion` and `\SetProtrusion` provide a similar interface for setting the character protrusion resp. expansion factors for a set of fonts.

`\SetProtrusion` This macro accepts three arguments: [options,] set of font characteristics and list of character protrusion factors.  
`\MT@pr@c@name`  
`\MT@load` A new macro called `\MT@pr@c@<name>` will be defined to be `<#3>` (i.e. the list of characters, not expanded).  
`\MT@extra@factor`

```

\MT@extra@preset 2236 \renewcommand*\SetProtrusion[2][]{%
\MT@extra@unit    2237 \let\MT@pr@c@name\@undefined
2238 \let\MT@load\@undefined
\MT@extra@context 2239 \let\MT@extra@factor\@undefined
2240 \let\MT@extra@unit\@undefined
2241 \let\MT@extra@preset\@undefined
2242 \let\MT@extra@context\@empty

```

Parse the optional first argument:

```
2243 \setkeys{MT@pr@c}{#1}%
```

If the user hasn't specified a name, we will create one.

```

2244 \MT@get@codes@name{pr}%
2245 \MT@set@pr@opt
2246 <debug>\MT@dinfo{1}{creating protrusion list '\MT@pr@c@name'}%
2247 \def\MT@permutelist{pr}%
2248 \setkeys{MT@pr@c}{#2}%

```

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

```
2249 \MT@permute
```

... which we can now define to be `<#3>`. We want the catcodes to be correct even if this is called in the preamble.

```

2250 \MT@begin@catcodes
2251 \MT@set@pr@list
2252 }

```

`\MT@set@pr@list` Here, as elsewhere, we have to make the definitions global, since they will occur inside a group.

```

2253 \def\MT@set@pr@list#1{%
2254 \global\MT@defn{MT@pr@c@\MT@pr@c@name}{#1}%
2255 \MT@end@catcodes
2256 }

```

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

```

\MT@load 2257 \renewcommand*\SetExpansion[2][]{%
\MT@extra@factor 2258 \let\MT@ex@c@name\@undefined
2259 \let\MT@load\@undefined
\MT@extra@stretch 2260 \let\MT@extra@factor\@undefined
\MT@extra@shrink 2261 \let\MT@extra@stretch\@undefined
\MT@extra@step 2262 \let\MT@extra@shrink\@undefined
2263 \let\MT@extra@step\@undefined
\MT@extra@auto 2264 \let\MT@extra@auto\@undefined
2265 \let\MT@extra@preset\@undefined
\MT@extra@context 2266 \let\MT@extra@context\@empty
2267 \setkeys{MT@ex@c}{#1}%

```

```

2268 \MT@get@codes@name{ex}%
2269 \MT@set@ex@opt
2270 (debug)\MT@info{1}{creating expansion list '\MT@ex@c@name'}%
2271 \def\MT@permutelist{ex@c}%
2272 \setkeys{MT@ex@c}{#2}%
2273 \MT@permute
2274 \MT@begin@catcodes
2275 \MT@set@ex@list
2276 }

\MT@set@ex@list Same story.
2277 \def\MT@set@ex@list#1{%
2278 \global\MT@defn{MT@ex@c@\MT@ex@c@name}{#1}%
2279 \MT@end@catcodes
2280 }

2281 (*beta)

\SetExtraSpacing
\MT@sp@c@name 2282 \renewcommand*\SetExtraSpacing[2][{}]{%
\MT@load 2283 \let\MT@sp@c@name\@undefined
2284 \let\MT@load\@undefined
\MT@extra@factor 2285 \let\MT@extra@factor\@undefined
\MT@extra@unit 2286 \let\MT@extra@unit\@undefined
\MT@extra@preset 2287 \let\MT@extra@preset\@undefined
2288 \let\MT@extra@context\@empty
\MT@extra@context 2289 \setkeys{MT@sp@c}{#1}%
2290 \MT@get@codes@name{sp}%
2291 \MT@set@sp@opt
2292 (debug)\MT@info{1}{creating space list '\MT@sp@c@name'}%
2293 \def\MT@permutelist{sp@c}%
2294 \setkeys{MT@sp@c}{#2}%
2295 \MT@permute
2296 \MT@begin@catcodes
2297 \MT@set@sp@list
2298 }

\MT@set@sp@list
2299 \def\MT@set@sp@list#1{%
2300 \global\MT@defn{MT@sp@c@\MT@sp@c@name}{#1}%
2301 \MT@end@catcodes
2302 }

\SetExtraKerning
\MT@kn@c@name 2303 \renewcommand*\SetExtraKerning[2][{}]{%
\MT@load 2304 \let\MT@kn@c@name\@undefined
2305 \let\MT@load\@undefined
\MT@extra@factor 2306 \let\MT@extra@factor\@undefined
\MT@extra@unit 2307 \let\MT@extra@unit\@undefined
\MT@extra@preset 2308 \let\MT@extra@preset\@undefined
2309 \let\MT@extra@context\@empty
\MT@extra@context 2310 \setkeys{MT@kn@c}{#1}%
2311 \MT@get@codes@name{kn}%
2312 \MT@set@kn@opt
2313 (debug)\MT@info{1}{creating kerning list '\MT@kn@c@name'}%
2314 \def\MT@permutelist{kn@c}%
2315 \setkeys{MT@kn@c}{#2}%
2316 \MT@permute
2317 \MT@begin@catcodes
2318 \MT@set@kn@list

```



```

2319 }

\MT@set@kn@list

2320 \def\MT@set@kn@list#1{%
2321   \global\MT@def@n{MT@kn@c@MT@kn@c@name}{#1}%
2322   \MT@end@catcodes
2323 }

2324 /beta

\MT@get@codes@name Use file name and line number as the list name if the user didn't bother creating
one.

2325 \def\MT@get@codes@name#1{%
2326   \MT@ifdefined@n@TF{MT@#1@c@name}{%
2327     \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
2328       \MT@warning{Redefining list '\@nameuse{MT@#1@c@name}'}%
2329     }%
2330   }%
2331   \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2332 }%
2333 \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%

Now that we know the name, we can cater for any set to be loaded by this list.

2334 \MT@ifdefined@c@T\MT@load{%
2335   \global\MT@let@nc{MT@#1@c@\MT@curr@set@name load}\MT@load
2336 }%
2337 }

\MT@set@pr@opt Three extra option for protrusion: factor, unit and preset.

2338 \def\MT@set@pr@opt{%
2339   \MT@set@opt@{pr}{factor}%
2340   \MT@set@opt@{pr}{unit}%
2341   \MT@set@opt@{pr}{preset}%
2342 }

\MT@set@ex@opt The extra options to \SetExpansion also have to be dealt with only after we know
the name.

2343 \def\MT@set@ex@opt{%
2344   \MT@ifdefined@c@T\MT@extra@factor{%
2345     \ifnum\MT@extra@factor>\@m
2346       \MT@warning@n1{Expansion factor \number\MT@extra@factor\space too
2347         large in list\MessageBreak '\MT@ex@c@name'. Setting it to the
2348         maximum of 1000}%
2349       \let\MT@extra@factor\@m
2350     \fi
2351     \global\MT@let@nc{MT@ex@c@\MT@ex@c@name @factor}\MT@extra@factor
2352   }%
2353   \MT@set@opt@{ex}{stretch}%
2354   \MT@set@opt@{ex}{shrink}%
2355   \MT@set@opt@{ex}{step}%
2356   \MT@set@opt@{ex}{auto}%
2357   \MT@set@opt@{ex}{preset}%
2358 }

2359 *beta

\MT@set@sp@opt

2360 \def\MT@set@sp@opt{%
2361   \MT@set@opt@{sp}{factor}%
2362   \MT@set@opt@{sp}{unit}%

```

```
2363 \MT@set@opt@{sp}{preset}%
2364 }
```

\MT@set@kn@opt

```
2365 \def\MT@set@kn@opt{%
2366 \MT@set@opt@{kn}{factor}%
2367 \MT@set@opt@{kn}{unit}%
2368 \MT@set@opt@{kn}{preset}%
2369 }
2370 /beta
```

\MT@set@opt@

```
2371 \def\MT@set@opt@#1#2{%
2372 \MT@ifdefined@n@T{MT@extra@#2}{%
2373 \global\MT@let@n@{MT@#1c@{csname MT@#1c@name\endcsname @#2}{MT@extra@#2}%
2374 }%
2375 }
```

\MT@define@code@key Define the keys for expansion and protrusion character code lists.

```
2376 \def\MT@define@code@key#1#2{%
2377 \define@key{MT@#2}{#1}[]{%
2378 \@tempcnta=\@ne
2379 \MT@map@clist@n{##1}{%
2380 \KV@sp@def\MT@val{####1}%
```

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

```
2381 \MT@get@highlevel{#1}%
2382 \MT@edef@n{MT@temp#1\romannumeral\@tempcnta}{\MT@val}%
2383 \advance\@tempcnta \@ne
2384 }%
2385 }%
2386 }
```

\MT@define@code@key@size

```
2387 \def\MT@define@code@key@size#1{%
2388 \define@key{MT@#1}{size}[]{%
2389 \MT@map@clist@n{##1}{%
2390 \KV@sp@def\MT@val{####1}%
2391 \expandafter\MT@get@range\MT@val--\@nil
2392 \ifx\MT@val\relax \else
2393 \expandafter\MT@xadd
2394 \csname MT@tempsize\endcsname
2395 {{{\MT@lower}{\MT@upper}{\csname MT@#1@name\endcsname}}}%
2396 \fi
2397 }%
2398 }%
2399 }
```

\MT@define@code@key@font

```
2400 \def\MT@define@code@key@font#1{%
2401 \define@key{MT@#1}{font}[]{%
2402 \MT@map@clist@n{##1}{%
2403 \KV@sp@def\MT@val{####1}%
2404 \expandafter\MT@get@font@and@size\MT@val////\@nil
2405 \global\MT@edef@n{MT@MT@permutelist @\@tempb}%
2406 {\csname MT@MT@permutelist @name\endcsname}%
2407 *debug
2408 \MT@info@n1{1}{initializing: use list for font \@tempb=\MT@val
2409 \ifx\MT@extra@context\empty\else\MessageBreak
```

```

2410                                     (context: \MT@extra@context)\fi}%
2411 /debug
2412     \expandafter\MT@xaddb
2413     \csname MT@\MT@permutelist @\@tempb @sizes\endcsname
2414     {{{\MT@val}{\m@ne}{\csname MT@#1@name\endcsname}}}%
2415     }%
2416 }%
2417 }

```

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

```

2418 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2419 \MT@ifempty{#1#2#3#4#5}\relax{%
2420 \let\@tempb\@empty
2421 \def\MT@temp{#1/#2/#3/#4/#5}%
2422 \MT@get@axis{encoding}{#1}%
2423 \MT@get@axis{family}{#2}%
2424 \MT@get@axis{series}{#3}%
2425 \MT@get@axis{shape}{#4}%

```

Append an asterisk for the size.

```

2426 \edef\@tempb{\@tempb*}%
2427 \MT@ifempty{#5}{%
2428 \MT@warn@axis@empty{size}{\string\normalsize}%
2429 \def\MT@val{*}%
2430 }{%
2431 \def\MT@val{#5}%
2432 }%
2433 \MT@get@size
2434 }%
2435 }

```

\MT@declare@codes

```

2436 \def\MT@declare@codes#1{%
2437 \define@key{MT@#1@c}{name}[]{}%
2438 \MT@ifempty{#1}\relax{%
2439 \MT@def@n{MT@#1@c@name}{##1}%
2440 }%
2441 }%
2442 \define@key{MT@#1@c}{load}[]{}%
2443 \MT@ifempty{#1}\relax{%
2444 \def\MT@load{##1}%
2445 }%
2446 }%
2447 \define@key{MT@#1@c}{factor}[]{}%
2448 \MT@ifempty{#1}\relax{%
2449 \def\MT@extra@factor{##1}%
2450 }%
2451 }%
2452 \MT@define@code@key{encoding}{#1@c}%
2453 \MT@define@code@key{family}{#1@c}%
2454 \MT@define@code@key{series}{#1@c}%
2455 \MT@define@code@key{shape}{#1@c}%
2456 \MT@define@code@key@size{#1@c}%
2457 \MT@define@code@key@font{#1@c}%
2458 \define@key{MT@#1@c}{preset}[]{}%
2459 \MT@ifempty{#1}\relax{%
2460 \def\MT@extra@preset{##1}%
2461 }%
2462 }%

```

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

```

2463 \define@key{MT@#1@c}{context}[]{%
2464 \MT@ifempty{##1}\relax{%
2465 \def\MT@extra@context{##1}%
2466 }%
2467 }%
2468 }

2469 \MT@declare@codes{pr}
2470 \MT@declare@codes{ex}
2471 \beta
2472 \MT@declare@codes{sp}
2473 \MT@declare@codes{kn}
2474 /\beta

```

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

```

2475 \define@key{MT@pr@c}{unit}[character]{%
2476 \let\MT@extra@unit\empty
2477 \KV@sp@def\@tempa{##1}%
2478 \MT@ifstreq\@tempa{relative}{%
2479 \MT@warning{Value 'relative' for key 'unit' is deprecated.\MessageBreak
2480 Use 'unit=character' instead. For now, I'll do it\MessageBreak
2481 for you}%
2482 \def\@tempa{character}%
2483 }\relax
2484 \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.

```

2485 \MT@ifdimen\@tempa{%
2486 \let\MT@extra@unit\@tempa
2487 }{%
2488 \MT@warning{'\@tempa' is not a dimension.\MessageBreak
2489 Ignoring it and setting values relative to\MessageBreak
2490 character widths}%
2491 }%
2492 }%
2493 }

```

`\MT@define@key@unit` Spacing and kerning codes may additionally be relative to space dimensions.

```

2494 \beta
2495 \def\MT@define@key@unit#1{%
2496 \define@key{MT@#1@c}{unit}[space]{%
2497 \let\MT@extra@unit\empty
2498 \KV@sp@def\@tempa{##1}%
2499 \MT@ifstreq\@tempa{relative}{%
2500 \MT@warning{Value 'relative' for key 'unit' is deprecated.\MessageBreak
2501 Use 'unit=character' instead. For now, I'll do it\MessageBreak
2502 for you}%
2503 \def\@tempa{character}%
2504 }\relax
2505 \MT@ifstreq\@tempa{character}\relax{%
2506 \let\MT@extra@unit\m@ne
2507 \MT@ifstreq\@tempa{space}\relax{%
2508 \MT@ifdimen\@tempa{%
2509 \let\MT@extra@unit\@tempa
2510 }{%
2511 \MT@warning{'\@tempa' is not a dimension.\MessageBreak
2512 Ignoring it and setting values relative to\MessageBreak
2513 width of space}%

```

```

2514         }%
2515     }%
2516 }%
2517 }%
2518 }

2519 \MT@define@key@unit{sp}
2520 \MT@define@key@unit{kn}
2521 \beta

```

`\MT@define@ex@key` The first argument to `\SetExpansion` accepts some more options.

```

2522 \def\MT@define@ex@key#1{%
2523   \define@key{MT@ex@}{#1}[]{}%
2524   \MT@ifempty{#1}\relax{}%
2525   \MT@ifnumber{#1}{%

```

A space terminates the number.

```

2526     \MT@def@n{MT@extra@#1}{##1 }%
2527   }{}%
2528   \MT@warning{%
2529     Value '##1' for option '#1' is not a number.\MessageBreak
2530     Ignoring it}%
2531 }{}%
2532 }{}%
2533 }{}%
2534 }

2535 \MT@define@ex@key{stretch}
2536 \MT@define@ex@key{shrink}
2537 \MT@define@ex@key{step}
2538 \define@key{MT@ex@}{auto}[true]{}%
2539 \KV@sp@def\@tempa{#1}%
2540 \csname if\@tempa\endcsname

```

Don't alter `\MT@extra@auto` for pdfTeX version older than 1.20.

```

2541 \MT@requires@pdftex4{%
2542   \def\MT@extra@auto{autoexpand}%
2543 }{}%
2544 \MT@warning{pdfTeX too old for automatic font expansion}%
2545 }
2546 \else
2547 \MT@requires@pdftex4{%
2548   \let\MT@extra@auto\@empty
2549 }\relax
2550 \fi
2551 }

```

### 13.5.4 Character Inheritance

`\DeclareCharacterInheritance` This macro may be used in the configuration files to declare characters that should inherit protrusion resp. expansion values from other characters. Thus, there is no need to define all accented characters (e. g. `\'a`, `\'a`, `\^a`, `\~a`, `\"a`, `\r{a}`, `\k{a}`, `\u{a}`), which will make the configuration files look much nicer and easier to maintain. If a single character of an inheritance list should have a different value, one can simply override it.

```

2552 \renewcommand*\DeclareCharacterInheritance[1]{}{}%
2553 \KV@sp@def\@tempa{#1}%

```

```

2554 \MT@begin@catcodes
2555 \MT@set@inh@list
2556 }

```

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

```

2557 \def\MT@set@inh@list#1#2{%
2558   \MT@ifempty\@tempa{%
2559     \MT@declare@char@inh{pr}{#1}{#2}%
2560     \MT@declare@char@inh{ex}{#1}{#2}%
2561     <beta>
2562     \MT@declare@char@inh{sp}{#1}{#2}%
2563     \MT@declare@char@inh{kn}{#1}{#2}%
2564     /beta>
2565   }{%
2566     \MT@map@clist@c\@tempa{%
2567       \KV@spdef\@tempa{#1}%
2568       \MT@ifempty\@tempa\relax{%
2569         \MT@is@feature{inheritance declaration for '#1'}{%
2570           \MT@exp@one@n\MT@declare@char@inh
2571             {\csname MT@rbba\@tempa\endcsname}{#1}{#2}%
2572         }%
2573       }%
2574     }%
2575   }%
2576 \MT@end@catcodes
2577 }

```

\MT@declare@char@inh The optional argument may be used to restrict the inheritance list to protrusion or expansion.

```

2578 \def\MT@declare@char@inh#1#2#3{%
2579   \MT@let@nc{MT@#1@inh@name}\@undefined
2580   \MT@get@inh@name{#1}%
2581   <debug> \MT@dinfo{1}{creating inheritance list '\@nameuse{MT@#1@inh@name}'}%
2582   \global\MT@def@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
2583   \def\MT@permutelist{#1@inh}%
2584   \setkeys{MT@#1@inh}{#2}%
2585   \MT@permute
2586 }

```

\MT@get@inh@name The inheritance lists cannot be named by the user.

```

2587 \def\MT@get@inh@name#1{%
2588   \MT@edef@n{MT@#1@inh@name}%
2589   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
2590 }

```

\MT@define@inh@key@encoding Parse the first argument. \DeclareCharacterInheritance may also be set up for various combinations.

```

2591 \def\MT@define@inh@key@encoding#1{%
2592   \define@key{MT@#1}{encoding}[]{%
2593     \def\MT@val{#1}%
2594     \expandafter\MT@encoding@check\MT@val,\@nil
2595     \MT@get@highlevel{encoding}%
2596     \MT@edef@n{MT@tempencoding\romannumeral1}{\MT@val}%
2597   }%
2598 }

```

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

```

2599 \def\MT@encoding@check#1,#2\@nil{%
2600   \MT@ifempty{#2}\relax{%

```

```

2601 \edef\MT@val{#1}%
2602 \MT@warning{You may only specify one encoding for character\MessageBreak
2603 inheritance lists. Ignoring encoding(s) #2}%
2604 }%
2605 }

```

\MT@define@inh@keys

```

2606 \def\MT@define@inh@keys#1{%
2607 \MT@define@inh@key@encoding{#1@inh}%

```

For the rest, we can reuse the key setup from \SetProtrusion resp. \SetExpansion.

```

2608 \MT@define@code@key{family}{#1@inh}%
2609 \MT@define@code@key{series}{#1@inh}%
2610 \MT@define@code@key{shape}{#1@inh}%
2611 \MT@define@code@key@size{#1@inh}%
2612 \MT@define@code@key@font{#1@inh}%
2613 }
2614 \MT@define@inh@keys{pr}
2615 \MT@define@inh@keys{ex}
2616 *beta
2617 \MT@define@inh@keys{sp}
2618 \MT@define@inh@keys{kn}
2619 /beta

```

\MT@inh@do Parse the second 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@pr@codes resp. \MT@set@ex@codes).

```

2620 \def\MT@inh@do#1,{%
2621 \ifx\relax#1\@empty \else
2622 \MT@inh@split #1==\relax
2623 \expandafter\MT@inh@do
2624 \fi
2625 }

```

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

```

2626 \def\MT@inh@split#1=#2=#3\relax{%
2627 \def\@tempa{#1}%
2628 \ifx\@tempa\@empty \else
2629 \MT@get@slot
2630 \ifnum\MT@char > \m@ne
2631 \let\MT@val\MT@char
2632 \MT@map@clist@n{#2}{%
2633 \def\@tempa{##1}%
2634 \ifx\@tempa\@empty \else
2635 \MT@get@slot
2636 \ifnum\MT@char > \m@ne
2637 \expandafter\MT@xadd
2638 \csname MT@inh@\MT@inh@name @\MT@val @\endcsname
2639 {{\MT@char}}}%
2640 \fi
2641 \fi
2642 }%
2643 *debug
2644 \MT@info@n1{2}{children of #1 (\MT@val):
2645 \nameuse{MT@inh@\MT@inh@name @\MT@val @}}%

```

```

2646 </debug>
2647   \fi
2648 \fi
2649 }

```

### 13.5.5 Permutation

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

```

2650 \def\MT@permute{%
2651   \let\MT@cnt@encoding\@ne
2652   \MT@permute@

```

Undefine commands for the next round.

```

2653   \MT@permute@reset
2654 }
2655 \def\MT@permute@{%
2656   \let\MT@cnt@family\@ne
2657   \MT@permute@@
2658   \MT@increment\MT@cnt@encoding
2659   \MT@ifdefined@n@T{\MT@tempencoding\romannumeral\MT@cnt@encoding}%
2660   \MT@permute@
2661 }
2662 \def\MT@permute@@{%
2663   \let\MT@cnt@series\@ne
2664   \MT@permute@@@
2665   \MT@increment\MT@cnt@family
2666   \MT@ifdefined@n@T{\MT@tempfamily\romannumeral\MT@cnt@family}%
2667   \MT@permute@@@
2668 }
2669 \def\MT@permute@@@{%
2670   \let\MT@cnt@shape\@ne
2671   \MT@permute@@@@
2672   \MT@increment\MT@cnt@series
2673   \MT@ifdefined@n@T{\MT@tempseries\romannumeral\MT@cnt@series}%
2674   \MT@permute@@@@
2675 }
2676 \def\MT@permute@@@@{%
2677   \MT@permute@@@@@
2678   \MT@increment\MT@cnt@shape
2679   \MT@ifdefined@n@T{\MT@tempshape\romannumeral\MT@cnt@shape}%
2680   \MT@permute@@@@@
2681 }

```

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

```

2682 \def\MT@permute@@@@@{%
2683   \MT@permute@define{encoding}%
2684   \ifMT@document
2685     \ifx\MT@tempencoding\@empty \else
2686       \MT@ifdefined@n@TF{T@MT@tempencoding}\relax
2687       {\expandafter\expandafter\expandafter\@gobble}%
2688     \fi

```



```

2689 \fi
2690 \MT@permute@@@@@
2691 }

\MT@permute@@@@@

2692 \def\MT@permute@@@@@{%
2693 \MT@permute@define{family}%
2694 \MT@permute@define{series}%
2695 \MT@permute@define{shape}%
2696 \edef\@tempa{\MT@tempencoding
2697             /\MT@tempfamily
2698             /\MT@tempseries
2699             /\MT@tempshape
2700             /\MT@ifdefined@c@T\MT@tempsize *}%

```

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

```

2701 \def\@tempb{////}%
2702 \ifx\@tempa\@tempb \else
2703 \ifx\MT@tempencoding\@empty
2704 \MT@warning{%
2705 You have to specify an encoding for\MessageBreak
2706 \nameuse{MT@abbr@MT@permutelist} list
2707 '\nameuse{MT@MT@permutelist @name}'.\MessageBreak
2708 Ignoring it}%
2709 \else
2710 \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

2711 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
2712 \MT@map@tlist@c
2713 \MT@tempsize
2714 \MT@check@rlist
2715 }%
2716 \expandafter\MT@xaddb
2717 \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
2718 \MT@tempsize
2719 <debug>
2720 \MT@dinfo@n{1}{initializing: use list for font \@tempa,\MessageBreak
2721 sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
2722 @sizes\endcsname}%
2723 </debug>
2724 }{%

```

Only one list should apply to a given combination.

```

2725 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
2726 \MT@warning{\nameuse{MT@abbr@MT@permutelist} list
2727 '\nameuse{MT@MT@permutelist @name}' will override list\MessageBreak
2728 '\nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}' for font '\@tempa'}%
2729 }%
2730 <debug>
2731 \MT@dinfo@n{1}{initializing: use list for font \@tempa
2732 \ifx\MT@extra@context\@empty\else\MessageBreak
2733 (context: \MT@extra@context)\fi}%
2734 </debug>
2735 }%
2736 \global\MT@edef@n{MT@MT@permutelist @\@tempa\MT@extra@context}%
2737 {\csname MT@MT@permutelist @name\endcsname}%
2738 \fi
2739 \fi

```

```

2740 }

\MT@permute@define Define the commands.
2741 \def\MT@permute@define#1{%
2742   \expandafter\@tempcnta=\csname MT@cnt@#1\endcsname\relax
2743   \MT@ifdefined@n@TF{MT@temp#1\romannumeral\@tempcnta}%
2744   {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\romannumeral\@tempcnta\endcsname}}%
2745   {\MT@let@nc{MT@temp#1}\@empty}%
2746 }

\MT@permute@reset Reset the commands.
2747 \def\MT@permute@reset{%
2748   \MT@permute@reset@{encoding}%
2749   \MT@permute@reset@{family}%
2750   \MT@permute@reset@{series}%
2751   \MT@permute@reset@{shape}%
2752   \let\MT@tempsize\undefined
2753 }

\MT@permute@reset@
2754 \def\MT@permute@reset@#1{%
2755   \@tempcnta=\@ne
2756   \MT@loop
2757   \MT@let@nc{MT@temp#1\romannumeral\@tempcnta}\@undefined
2758   \advance\@tempcnta\@ne
2759   \MT@ifdefined@n@TF{MT@temp#1\romannumeral\@tempcnta}%
2760   \iftrue
2761   \iffalse
2762   \MT@repeat
2763 }

\MT@check@rlist For every new range item in \MT@tempsize, check whether it overlaps with ranges
in the existing list.
2764 \def\MT@check@rlist#1{%
2765   \expandafter\MT@check@rlist@#1%
2766 }

\MT@check@rlist@ Define the current new range and ...
2767 \def\MT@check@rlist@#1#2#3{%
2768   \def\@tempb{#1}%
2769   \def\@tempc{#2}%
2770   \@tempswafalse
2771   \expandafter\MT@map@tlist@
2772   \csname MT@\MT@permutelist @\@tempa\MT@extra@context @size\endcsname
2773   \MT@check@range
2774 }

\MT@check@range ... recurse through the list of existing ranges.
2775 \def\MT@check@range#1{%
2776   \expandafter\MT@check@range@#1%
2777 }

\MT@check@range@ \@tempb and \@tempc are lower resp. upper bound of the new range, <#2> and
<#3> those of the existing range.
2778 \def\MT@check@range@#1#2#3{%
2779   \MT@ifeq{#2}\m@ne{%
2780     \MT@ifeq\@tempc\m@ne{%

```

- Both items are simple sizes.

```
2781      \MT@ifeq\@tempb{#1}\@tempwattrue\relax
2782      }{%
```

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

```
2783      \MT@ifgt\@tempb{#1}\relax{%
2784      \MT@ifgt\@tempc{#1}{%
2785      \@tempwattrue
2786      \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
2787      }\relax
2788      }%
2789      }%
2790      }{%
2791      \MT@ifeq\@tempc\m@ne{%
```

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

```
2792      \MT@iflt\@tempb{#2}{%
2793      \MT@iflt\@tempb{#1}\relax\@tempwattrue
2794      }\relax
2795      }{%
```

- Both items are ranges.

```
2796      \MT@iflt\@tempb{#2}{%
2797      \MT@ifgt\@tempc{#1}{%
2798      \@tempwattrue
2799      \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
2800      }\relax
2801      }\relax
2802      }%
2803      }%
2804      \if@tempwa
2805      \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
2806      '\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
2807      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.

```
2808      \expandafter\MT@tlist@break
2809      \fi
2810      }
```

## 13.6 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` and `activate`, and `spacing` and `kerning`. Specifying font sets is not allowed.

```
2811 \def\microtypesetup{\setkeys{MT}}
2812 \MT@addto@setup{\def\microtypesetup{\setkeys{MTX}}}
2813 \def\MT@define@optionX#1#2{%
2814   \define@key{MTX}{#1}[true]{%
2815     \KV@sp@def\@tempb{#1}%
2816     \MT@map@clist@n{##1}{%
2817       \KV@sp@def\MT@val{####1}%
```

```

2818 \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
2819 \MT@ifempty\MT@val\relax{%
2820 \@tempcnta=\m@ne
2821 \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.

```

2822 \MT@checksetup\@tempb{%
2823 \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
2824 \MT@info{Enabling #1
2825 (level \number\csname MT@\@tempb @level\endcsname)}}%
2826 }%
2827 }{%
2828 \MT@ifstreq\MT@val{false}{%
2829 \@tempcnta=\z@
2830 \MT@info{Disabling #1}%
2831 }{%
2832 \MT@ifstreq\MT@val{compatibility}{%
2833 \MT@checksetup\@tempb{%
2834 \@tempcnta=\@ne
2835 \MT@let\nc{MT@\@tempb @level}\@ne
2836 \MT@info{Setting #1 to level 1}%
2837 }%
2838 }{%
2839 \MT@ifstreq\MT@val{nocompatibility}{%
2840 \MT@checksetup\@tempb{%
2841 \@tempcnta=\tw@
2842 \MT@let\nc{MT@\@tempb @level}\tw@
2843 \MT@info{Setting #1 to level 2}%
2844 }%
2845 }{%
2846 \MT@warning{%
2847 Value '\MT@val' for key '#1' not recognized.\MessageBreak
2848 Use any of 'true', 'false', 'compatibility' or\MessageBreak
2849 'nocompatibility'}%
2850 }%
2851 }%
2852 }%
2853 }%
2854 \ifnum\@tempcnta>\m@ne
2855 #2\@tempcnta\relax
2856 \fi
2857 }%
2858 }%
2859 }%
2860 }

```

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

```

2861 \def\MT@checksetup#1{%
2862 \expandafter\csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
2863 \expandafter\@firstofone
2864 \else
2865 \MT@warning{%
2866 You cannot enable \@nameuse{MT@abbr@#1} if it was disabled\MessageBreak
2867 in the package options,}%
2868 \expandafter\@gobble
2869 \fi
2870 }

```

```

2871 \MT@define@optionX{protrusion}\pdfprotrudechars
2872 \MT@define@optionX{expansion}\pdfadjustspacing
2873 (\beta)

```

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

```

2874 \def\MT@define@optionX@#1#2{%
2875   \define@key{MTX}{#1}[true]{%
2876     \KV@sp@def\@tempb{#1}%
2877     \MT@map@clist@n{##1}{%
2878       \KV@sp@def\MT@val{###1}%
2879       \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
2880       \MT@ifempty\MT@val\relax{%
2881         \@tempcnta=\m@ne
2882         \MT@ifstreq\MT@val{true}{%
2883           \MT@checksetup\@tempb{%
2884             \@tempcnta=\@ne
2885             \MT@info{Enabling #1}%
2886           }%
2887         }%
2888         \MT@ifstreq\MT@val{false}{%
2889           \@tempcnta=\z@
2890           \MT@info{Disabling #1}%
2891         }%
2892         \MT@warning{%
2893           Value '\MT@val' for key '#1' not recognized.\MessageBreak
2894           Use either 'true' or 'false'}%
2895       }%
2896     }%
2897     \ifnum\@tempcnta>\m@ne
2898       #2\relax
2899     \fi
2900   }%
2901 }%
2902 }%
2903 }

2904 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
2905 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
2906   \pdfappendkern \@tempcnta}
2907 (\beta)
2908 \define@key{MTX}{activate}[]{%
2909   \setkeys{MTX}{protrusion={#1}}%
2910   \setkeys{MTX}{expansion={#1}}%
2911 }

```

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

```

2912 \def\MT@gobblethree#1#2#3{}
2913 \let\MT@saved@setupfont\MT@setupfont
2914 \define@key{MTX}{disable}[]{%
2915   \MT@info{Inactivate '\MT@MT' package}%
2916   \let\MT@setupfont\MT@gobblethree
2917 }
2918 \define@key{MTX}{enable}[]{%
2919   \MT@info{Reactivate '\MT@MT' package}%
2920   \let\MT@setupfont\MT@saved@setupfont
2921 }

```

## 13.7 Package Options

### 13.7.1 Declaring the Options

```

\ifMT@opt@expansion Keep track of whether the user explicitly set these options.
\ifMT@opt@auto 2922 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 2923 \newif\ifMT@opt@auto
                2924 \newif\ifMT@opt@DVI

\MT@define@option expansion and protrusion may be true, false, compatibility, nocompatibility
                and/or a <set name>.
2925 \def\MT@define@option#1{%
2926   \define@key{MT}{#1}[true]{%
2927     \csname MT@opt@#1true\endcsname
2928     \MT@map@cliston{##1}{%
2929       \KV@sp@def\MT@val{###1}%
2930       \MT@ifempty\MT@val\relax{%
2931         \csname MT@#1true\endcsname
2932         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2933         \MT@ifstreq\MT@val{true}\relax
2934         {%
2935           \MT@ifstreq\MT@val{false}{%
2936             \csname MT@#1false\endcsname
2937           }{%
2938             \MT@ifstreq\MT@val{compatibility}{%
2939               \MT@let@nc{MT@\@tempb @level}\@ne
2940             }{%
2941               \MT@ifstreq\MT@val{nocompatibility}{%
2942                 \MT@let@nc{MT@\@tempb @level}\tw@
2943               }{%
2944                 \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
2945                   \global\MT@edef@n{MT@\@tempb @setname}{\MT@val}%
2946                 }{%
2947                   \global\MT@edef@n{MT@\@tempb @setname}%
2948                     {\@nameuse{MT@default@\@tempb @set}}%
2949                   \MT@warning@n{%
2950                     The #1 set '\MT@val' is undeclared.\MessageBreak
2951                     Using set '\@nameuse{MT@\@tempb @setname}' instead}%
2952                   }%
2953                 }%
2954               }%
2955             }%
2956           }%
2957         }%
2958       }%
2959     }%
2960   }

2961 \MT@define@option{protrusion}
2962 \MT@define@option{expansion}

activate is a shortcut for protrusion and expansion (and spacing?).
2963 \define@key{MT}{activate}[]{%
2964   \setkeys{MT}{protrusion={#1}}%
2965   \setkeys{MT}{expansion={#1}}%
2966 }
2967 <beta>

```

`\MT@define@option@` spacing and kerning do not have a compatibility level.

```

2968 \def\MT@define@option@#1{%
2969   \define@key{MT}{#1}[true]{%
2970     \csname MT@opt@#1true\endcsname
2971     \MT@map@clist@n{##1}{%
2972       \KV@sp@def\MT@val{###1}%
2973       \MT@ifempty\MT@val\relax{%
2974         \csname MT@#1true\endcsname
2975         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2976         \MT@ifstreq\MT@val{true}\relax
2977         {%
2978           \MT@ifstreq\MT@val{false}{%
2979             \csname MT@#1false\endcsname
2980             }{%
2981               \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
2982                 \global\MT@edef@n{MT@\@tempb @setname}{\MT@val}%
2983                 }{%
2984                   \global\MT@edef@n{MT@\@tempb @setname}%
2985                   {\@nameuse{MT@default@\@tempb @set}}%
2986                   \MT@warning@n{%
2987                     The #1 set '\MT@val' is undeclared.\MessageBreak
2988                     Using set '\@nameuse{MT@\@tempb @setname}' instead}%
2989                   }%
2990                 }%
2991               }%
2992             }%
2993           }%
2994         }%
2995       }
2996 \MT@define@option@{spacing}
2997 \MT@define@option@{kerning}
2998 </beta>

```

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

```

2999 \def\MT@def@bool@opt#1#2{%
3000   \define@key{MT}{#1}[]{%
3001     \MT@ifempty{##1}%
3002     {\def\@tempa{true}}%
3003     {\def\@tempa{##1}}%
3004     \MT@ifstreq\@tempa{true}\relax{%
3005       \MT@ifstreq\@tempa{false}\relax{%
3006         \MT@warning@n{%
3007           '##1' is not an admissible value for option\MessageBreak
3008           '#1'. Assuming 'false'}%
3009         \def\@tempa{false}%
3010       }%
3011     }%
3012     #2%
3013   }%
3014 }

```

`\MT@def@simple@bool@opt` Boolean options that only set the switch.

```

3015 \def\MT@def@simple@bool@opt#1{\MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
3016 \MT@map@tlist@n{{draft}{auto}{selected}}%
3017 <beta>{babel}%
3018 }\MT@def@simple@bool@opt

```

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

```

3019 \MT@def@bool@opt{DVIoutput}{%
3020   \csname if\@tempa\endcsname
3021   \ifnum\pdfoutput>\z@ \MT@opt@DVIttrue\fi
3022   \pdfoutput\z@
3023   \else
3024   \ifnum\pdfoutput<\@ne \MT@opt@DVIttrue \fi
3025   \pdfoutput\@ne
3026   \fi
3027 }
```

Setting the `defersetup` option to false will restore the old behaviour, where the setup took place at the time when the package was loaded. This is undocumented, since I would like to know whether it is actually 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.

```

3028 \MT@def@bool@opt{defersetup}{%
3029   \csname if\@tempa\endcsname \else
3030   \AtEndOfPackage{%
3031     \MT@setup@microtype
3032     \let\MT@setup@microtype\@empty
3033     \let\MT@addto@setup\@firstofone
3034   }%
3035   \fi
3036 }
```

`final` is the opposite to `draft`.

```

3037 \MT@def@bool@opt{final}{%
3038   \csname if\@tempa\endcsname
3039   \MT@draftfalse
3040   \else
3041   \MT@drafttrue
3042   \fi
3043 }
```

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

```

3044 \define@key{MT}{verbose}[]{}%
3045   \let\MT@vinfo\MT@info@nl
3046   \MT@ifempty{#1}%
3047   {\def\@tempa{true}}%
3048   {\def\@tempa{#1}}%
3049   \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```

3050   \MT@ifstreq\@tempa{errors}{%
3051     \let\MT@warning\MT@warn@err
3052     \let\MT@warning@nl\MT@warn@err
3053   }{%
3054     \let\MT@vinfo\@gobble
3055     \MT@ifstreq\@tempa{false}\relax{%
3056       \MT@warning@nl{%
3057         '#1' is not an admissible value for option\MessageBreak
3058         'verbose'. Assuming 'false'}%
3059     }%
3060   }%
3061 }
```



3062 }

\MT@def@num@opt Options with numerical keys: factor, stretch, shrink, step, letterspacing.

```
3063 \def\MT@def@num@opt#1{%
3064   \define@key{MT}{#1}[] {%
3065     \MT@ifempty{#1}%
3066     {\MT@let@cn\@tempa{MT@#1@default}}%
3067     {\def\@tempa{#1 }}%

```

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

```
3068   \MT@ifnumber\@tempa{%
3069     \MT@edef@n{MT@#1}{\@tempa}%
3070     }{\MT@warning@n1{%
3071       Value '#1' for option '#1' is not a number.\MessageBreak
3072       Using default value of \number\@nameuse{MT@#1@default}}%
3073     }%
3074   }%
3075 }

3076 \MT@map@tlist@n{{stretch}{shrink}{step}%
3077 <beta>{letterspacing}%
3078 }\MT@def@num@opt

```

factor will define the protrusion factor only.

```
3079 \define@key{MT}{factor}[] {%
3080   \MT@ifempty{#1}%
3081   {\let\@tempa\MT@factor@default}%
3082   {\def\@tempa{#1 }}%
3083   \MT@ifnumber\@tempa{%
3084     \MT@edef@n{MT@pr@factor}{\@tempa}%
3085     }{\MT@warning@n1{%
3086       Value '#1' for option 'factor' is not a number.\MessageBreak
3087       Using default value of \number\MT@factor@default}%
3088     }%
3089 }

```

Unit for codes.

```
3090 \define@key{MT}{unit}[] {%
3091   \MT@ifempty{#1}%
3092   {\def\@tempa{character}}%
3093   {\KV@sp@def\@tempa{#1}}%
3094   \MT@ifstreq\@tempa{relative}%
3095   \MT@warning{Value 'relative' for option 'unit' is deprecated.\MessageBreak
3096     Use 'unit=character' instead. For now, I'll do it.\MessageBreak
3097     for you}%
3098   \def\@tempa{character}%
3099   }\relax
3100 \MT@ifstreq\@tempa{character}\relax{%
3101   \MT@ifdimen\@tempa{%
3102     \let\MT@pr@unit\@tempa
3103   }{%
3104     \MT@warning@n1{'\@tempa' is not a dimension. Ignoring it and.\MessageBreak
3105       setting values relative to character widths}%
3106   }%
3107 }%
3108 }

```

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 pdf<sub>T</sub><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 pdf<sub>T</sub><sub>E</sub>X.)

```
3109 \MT@protrusiontrue
3110 \ifnum\pdfoutput=\z@ \else
```

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

```
3111 \MT@requires@pdftex4{%
3112 \MT@expansiontrue
3113 \MT@autottrue
3114 }\relax
3115 \fi
```

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

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

```
3116 \define@key{MT}{config}[]{\relax}
3117 \def\MT@get@config#1config=#2,#3\@nil{%
3118 \MT@ifempty{#2}%
3119 {\def\MT@config@file{\MT@MT.cfg}}%
3120 {\KV@sp@def\MT@config@file{#2.cfg}}%
3121 }
3122 \expandafter\expandafter\expandafter\MT@get@config
3123 \csname opt@%currname.%current\endcsname,config=,\@nil
```

Load the file.

```
3124 \IfFileExists{\MT@config@file}{%
3125 \MT@info@nl{Loading configuration file \MT@config@file}%
3126 \MT@begin@catcodes
3127 \let\MT@begin@catcodes\relax
3128 \let\MT@end@catcodes\relax
3129 \let\MT@curr@file\MT@config@file
3130 \input{\MT@config@file}%
3131 \endgroup
3132 }{%
3133 \MT@warning@nl{%
3134 Could not find configuration file '\MT@config@file'!\MessageBreak
3135 This will almost certainly cause undesired results.\MessageBreak
3136 Please fix your installation}%
3137 }
```

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

```
3138 \MT@ifdefined@c@TF\MT@default@pr@set\relax{\gdef\MT@default@pr@set{all}}
3139 \MT@ifdefined@c@TF\MT@default@ex@set\relax{\gdef\MT@default@ex@set{all}}
3140 <beta>
3141 \MT@ifdefined@c@TF\MT@default@sp@set\relax{\gdef\MT@default@sp@set{all}}
3142 \MT@ifdefined@c@TF\MT@default@kn@set\relax{\gdef\MT@default@kn@set{all}}
3143 </beta>
```

### 13.7.2 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 has not been 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 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}}
\ifundefined{Microtype@Hook}
{\let\Microtype@Hook\MinionPro@MT@Hook}
{\g@addto@macro\Microtype@Hook{\MinionPro@MT@Hook}}
```

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

```
3144 \MT@ifdefined@c@T\MicroType@Hook{%
3145   \MT@warning@n1{%
3146     Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3147     Use \string\Microtype@Hook\space instead}\MicroType@Hook}
3148 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook
```

### 13.7.3 Processing the Options

`\MT@ProcessOptionsWithKV` Parse options.

```
3149 \def\MT@ProcessOptionsWithKV#1{%
3150   \let\@tempc\relax
3151   \let\KV@tempa\@empty
3152   \MT@map@clist@c\@classoptionslist{%
3153     \def\CurrentOption{##1}%
3154     \MT@ifdefined@n@T{KV@#1\CurrentOption}{%
3155       \edef\KV@tempa{\KV@tempa,\CurrentOption,%
3156         \expandtwoargs\removeelement\CurrentOption
3157         \@unusedoptionlist\@unusedoptionlist
3158       }%
3159     }%
3160     \edef\KV@tempa{%
3161       \noexpand\setkeys{#1}{%
3162         \KV@tempa\@optionlist{\@currname.\@current}%
3163       }%
3164     }%
3165     \KV@tempa
3166     \AtEndOfPackage{\let\@unprocessedoptions\relax}%
3167     \let\CurrentOption\@empty
3168   }
3169 \MT@ProcessOptionsWithKV{MT}
```

Now we can take the appropriate actions. We also tell the log file which options the user has chosen (in case it's interested).

```
3170 \MT@addto@setup{%
3171 \ifMT@draft
```

We disable most of what we've just defined in the 3171 lines above if we are running in draft mode.

```

3172 \MT@warning@n1{'draft' option active.\MessageBreak
3173           Disabling all micro-typographic extensions.\MessageBreak
3174           This might lead to different line and page breaks}
3175 \MT@protrusionfalse
3176 \MT@expansionfalse
3177 <beta>
3178 \MT@spacingfalse
3179 \MT@kerningfalse
3180 \MT@babelfalse
3181 </beta>
3182 \let\MT@setupfont\relax
3183 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
3184 \renewcommand*{\UseMicrotypeSet[2] [] {}
3185 \renewcommand*{\SetProtrusion[3] [] {}
3186 \renewcommand*{\SetExpansion[3] [] {}
3187 <beta>
3188 \renewcommand*{\SetExtraSpacing[3] [] {}
3189 \renewcommand*{\SetExtraKerning[3] [] {}
3190 </beta>
3191 \renewcommand*{\DeclareCharacterInheritance[3] [] {}
3192 \renewcommand*{\DeclareMicrotypeAlias[2] {}
3193 \renewcommand*{\LoadMicrotypeFile[1] {}
3194 \renewcommand*{\microtypesetup[1] {}
3195 \renewcommand*{\microtypecontext[1] {}
3196 \else

```

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

```

3197 \ifnum\pdfoutput=\z@
3198   \ifMT@opt@expansion \else
3199     \MT@expansionfalse
3200   \fi
3201 \fi

```

pdfTeX can create DVI output, too. However, both the DVI viewer and dvips need to find actual fonts. Therefore, expansion will only work if the fonts for different degrees of expansion are readily available.

Some packages depend on the value of \pdfoutput and will get confused if it is changed after they have been loaded. These packages are, among others: color, graphics, hyperref, crop, contour, pstricks and, as a matter of course, ifpdf. Instead of testing for each package (that's not our job), we only say that it was microtype that changed it. This must be sufficient!

```

3202 \MT@info@n1{Generating \ifnum\pdfoutput=\z@ DVI \else PDF \fi output%
3203           \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%

```

Protrusion.

```

3204 \ifMT@protrusion
3205   \edef\MT@active@features{\MT@active@features,pr}
3206   \pdfprotrudechars\MT@pr@level
3207   \MT@info@n1{Character protrusion enabled (level \number\MT@pr@level)%
3208             \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3209             factor: \number\MT@pr@factor\fi
3210             \ifx\MT@pr@unit\empty \else,\MessageBreak unit: \MT@pr@unit\fi}

```

We have to make sure that font sets are active. If the user didn't activate any, we use those sets declared by `\DeclareMicrotypeSetDefault`.

```

3211 \MT@ifdefined@c@TF\MT@pr@setname{%
3212 \MT@info@nl{Using protrusion set '\MT@pr@setname'}%
3213 }{%
3214 \global\let\MT@pr@setname\MT@default@pr@set
3215 \MT@info@nl{Using default protrusion set '\MT@pr@setname'}%
3216 }
3217 \else
3218 \let\MT@protrusion\relax
3219 \MT@info@nl{No character protrusion}
3220 \fi

```

Expansion.

```

3221 \ifMT@expansion

```

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

```

3222 \ifnum\MT@stretch=\m@ne
3223 \let\MT@stretch\MT@stretch@default
3224 \fi

```

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

```

3225 \ifnum\MT@shrink=\m@ne
3226 \ifnum\MT@stretch>\z@
3227 \let\MT@shrink\MT@stretch
3228 \else
3229 \let\MT@shrink\MT@shrink@default
3230 \fi
3231 \fi

```

If step has not been specified, we will set it to  $\min(\text{stretch}, \text{shrink})/5$ , rounded off, minimum value 1.

```

3232 \ifnum\MT@step=\m@ne
3233 \ifnum\MT@stretch>\MT@shrink
3234 \ifnum\MT@shrink=\z@
3235 \@tempcnta=\MT@stretch
3236 \else
3237 \@tempcnta=\MT@shrink
3238 \fi
3239 \else
3240 \ifnum\MT@stretch=\z@
3241 \@tempcnta=\MT@shrink
3242 \else
3243 \@tempcnta=\MT@stretch
3244 \fi
3245 \fi
3246 \divide\@tempcnta 5\relax
3247 \else
3248 \@tempcnta=\MT@step
3249 \ifnum\@tempcnta=\z@
3250 \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
3251 Setting it to one}
3252 \fi
3253 \fi
3254 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3255 \edef\MT@step{\number\@tempcnta\space}

```

`\MT@auto` Automatic expansion of the font? This new feature of pdf<sub>T</sub><sub>E</sub>X 1.20 makes the *hz-*

algorithm really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdf $\TeX$ ).

```
3256 \let\MT@auto\@empty
3257 \ifMT@auto
3258 \MT@requires@pdftex4{%
```

We turn off automatic expansion if output mode is DVI.

```
3259 \ifnum\pdfoutput=\z@
3260 \ifMT@opt@auto
3261 \MT@warning@nl{%
3262 Automatic font expansion only works for PDF output.\MessageBreak
3263 However, you are creating a DVI file. I will switch\MessageBreak
3264 automatic font expansion off and hope that expanded\MessageBreak
3265 fonts are available}
3266 \fi
3267 \MT@autofalse
3268 \else
3269 \def\MT@auto{autoexpand}
3270 \fi
```

Also, if pdf $\TeX$  is too old.

```
3271 }{%
3272 \ifMT@opt@auto
3273 \MT@warning@nl{%
3274 The pdftex you are using is too old for automatic\MessageBreak
3275 font expansion. I will switch it off and hope that\MessageBreak
3276 expanded fonts are available on your system.\MessageBreak
3277 Install pdftex version 1.20 or newer}
3278 \fi
3279 \MT@autofalse
3280 \def\MT@auto{1000 }
3281 }
```

No automatic expansion.

```
3282 \else
3283 \ifnum\MT@pdftex@no < 4
3284 \def\MT@auto{1000 }
3285 \fi
3286 \fi
```

Choose the appropriate macro for selected expansion.

```
3287 \ifMT@selected
3288 \let\MT@set@ex@codes\MT@set@ex@codes@s
3289 \else
3290 \let\MT@set@ex@codes\MT@set@ex@codes@n
3291 \fi
```

Filter out stretch=0, shrink=0, since it would result in an pdf $\TeX$  error.

```
3292 \ifnum\MT@stretch=\z@
3293 \ifnum\MT@shrink=\z@
3294 \MT@warning@nl{%
3295 Both the stretch and shrink limit are set to zero.\MessageBreak
3296 Disabling font expansion}
3297 \MT@expansionfalse
3298 \fi
3299 \fi
3300 \fi
3301 \ifMT@expansion
3302 \edef\MT@active@features{\MT@active@features,ex}%
```

```

3303 \pdfadjustspacing\MT@ex@level
3304 \MT@info@n1{\ifMT@auto\else Non-\fi Automatic font expansion enabled
3305         (level \number\MT@ex@level),\MessageBreak
3306         stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3307         step: \number\MT@step, \ifMT@selected\else non-\fi selected}
3308 \MT@ifdefined@c@TF\MT@ex@setname{%
3309 \MT@info@n1{Using expansion set '\MT@ex@setname'}%
3310 }{%
3311 \global\let\MT@ex@setname\MT@default@ex@set
3312 \MT@info@n1{Using default expansion set '\MT@ex@setname'}%
3313 }

```

Inside `\showhyphens`, font expansion should be disabled.

```

3314 \CheckCommand*{\showhyphens}[1]{%
3315 \setbox0\vbox{\color@begingroup\everypar{}\parfillskip\z@skip
3316 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3317 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

```

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

```

3318 \gdef\showhyphens#1{%
3319 \setbox0\vbox{%
3320 \color@begingroup
3321 \pdfadjustspacing\z@
3322 \everypar{%
3323 \parfillskip\z@skip\hsize\maxdimen
3324 \normalfont
3325 \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ #1%
3326 \color@endgroup}}
3327 \else
3328 \let\MT@expansion\relax
3329 \MT@info@n1{No font expansion}
3330 \fi
3331 \fi
3332 }
3333 *beta
3334 \MT@requires@pdfTeX6{%
3335 \MT@addto@setup{%

```

Spacing.

```

3336 \ifMT@spacing
3337 \edef\MT@active@features{\MT@active@features,sp}
3338 \pdfadjustinterwordglue\@ne
3339 \MT@info@n1{Adjustment of interword spacing enabled}
3340 \MT@ifdefined@c@TF\MT@sp@setname{%
3341 \MT@info@n1{Using spacing set '\MT@sp@setname'}%
3342 }{%
3343 \global\let\MT@sp@setname\MT@default@sp@set
3344 \MT@info@n1{Using default spacing set '\MT@sp@setname'}%
3345 }

```

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 this message from Philipp Lehman: [ddtbaj\\$rob\\$1@online.de](mailto:ddtbaj$rob$1@online.de) on August 16, 2005.

```

3346 \ifnum\sfcode'\. > 1500
3347 \ifMT@babel \else
3348 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%

```

```

3349         \MT@warning@nl{%
3350             \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3351             interword spacing will disable it. You might want\MessageBreak
3352             to add '\MT@MT context{spacing=nonfrench}'\MessageBreak
3353             to your preamble}%
3354     }%
3355     \fi
3356 \fi
3357 \else
3358     \let\MT@spacing\relax
3359     \MT@info@nl{No adjustment of interword spacing}
3360 \fi

```

Kerning is always active (because of letterspacing). Hence, we also don't set `\MT@kerning` to `\relax`.

```

3361 \edef\MT@active@features{\MT@active@features,kn}
3362 \ifMT@kerning
3363     \pdfprependkern\@ne
3364     \pdfappendkern\@ne
3365     \MT@info@nl{Adjustment of character kerning enabled}
3366     \MT@ifdefined@c@TF\MT@kn@setname{%
3367         \MT@info@nl{Using kerning set '\MT@kn@setname'}%
3368     }{%
3369         \global\let\MT@kn@setname\MT@default@kn@set
3370         \MT@info@nl{Using default kerning set '\MT@kn@setname'}%
3371     }
3372 \else
3373     \MT@info@nl{No adjustment of character kerning}
3374 \fi
3375 \ifnum\MT@letterspacing=\m@ne
3376     \let\MT@letterspacing\MT@letterspacing@default
3377 \fi
3378 }

```

If pdf<sub>T</sub><sub>E</sub>X is too old, we disable spacing and kerning.

```

3379 }{%
3380     \MT@addto@setup{%
3381         \ifMT@spacing
3382             \MT@warning@nl{Adjustment of interword spacing only works with\MessageBreak
3383                 pdftex version 1.4x or newer. Switching it off}%
3384         \else
3385             \MT@info@nl{No adjustment of interword spacing}
3386         \fi
3387         \MT@spacingfalse
3388         \let\MT@spacing\relax
3389         \ifMT@kerning
3390             \MT@warning@nl{Character kerning only works with\MessageBreak
3391                 pdftex version 1.4x or newer. Switching it off}%
3392         \else
3393             \MT@info@nl{No adjustment of character kerning}
3394         \fi
3395         \MT@kerningfalse
3396         \let\MT@kerning\relax
3397     }
3398 }

```

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

```

3399 \MT@addto@setup{%

```



```

3400 \ifMT@babel
3401 \@ifpackageloaded{babel}{%
3402   \MT@info@nl{Redefining babel's language switching commands}
3403   \let\MT@orig@select@language\select@language
3404   \def\select@language#1{%
3405     \MT@orig@select@language{#1}%
3406     \MT@ifdefined@n@TF{MT@babel@#1}{%
3407       \MT@vinfo{Changing to language '#1' on line \the\inputlineno}%
3408       \expandafter\MT@exp@one@n\expandafter\microtypecontext
3409       \csname MT@babel@#1\endcsname
3410     }{%
3411       \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3412     }%
3413   }
3414   \let\MT@orig@foreign@language\foreign@language
3415   \def\foreign@language#1{%
3416     \MT@orig@foreign@language{#1}%
3417     \MT@ifdefined@n@TF{MT@babel@#1}{%
3418       \MT@vinfo{Changing to context '#1' on line \the\inputlineno}%
3419       \expandafter\MT@exp@one@n\expandafter\microtypecontext
3420       \csname MT@babel@#1\endcsname
3421     }{%
3422       \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3423     }%
3424   }

```

Disable babel's active characters.

```

3425 \ifMT@kerning
3426 \@tempswafalse
3427 \@ifpackagewith{babel}{french}\@tempswattrue\relax
3428 \@ifpackagewith{babel}{frenchb}\@tempswattrue\relax
3429 \@ifpackagewith{babel}{francais}\@tempswattrue\relax
3430 \if@tempswa
3431 \NoAutoSpaceBeforeFDP
3432 \MT@warning@nl{Switching off French babel's active punctuation characters}%
3433 \fi
3434 \fi
3435 }{%
3436 \MT@warning@nl{You did not load the babel package.\MessageBreak
3437   The 'babel' option won't have any effect}
3438 }
3439 \fi
3440 }
3441 /beta

```

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

```

3442 \MT@addto@setup{%
3443   \ifx\MT@active@features\@empty \else
3444     \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3445   \fi
3446   \MT@documenttrue
3447 }

```

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.

```

3448 \AtBeginDocument\selectfont

```

This is the current file.

```
3449 \edef\MT@curr@file{\jobname.tex}
3450 \end{package}
```

That was that.

## 14 Configuration Files

Let's now write the font configuration files.

```
3451 \begin{fontconfig}
3452
```

### 14.1 Font Sets

We first declare some sets in the main configuration file.

```
3453 \begin{fontconfig}
3454 %%% -----
3455 %%% FONT SETS
3456
3457 \DeclareMicrotypeSet{all}
3458 { }
3459
3460 \DeclareMicrotypeSet{allmath}
3461 { encoding = {T1,LY1,OT1,OT4,T5,TS1,OML,OMS,U} }
3462
3463 \DeclareMicrotypeSet{alltext}
3464 { encoding = {T1,LY1,OT1,OT4,T5,TS1} }
3465
3466 \DeclareMicrotypeSet{basemath}
3467 { encoding = {T1,LY1,OT1,OT4,T5,OML,OMS},
3468   family = {rm*,sf*},
3469   series = {m},
3470   size = {normalsize,footnotesize,small,large}
3471 }
3472
3473 \DeclareMicrotypeSet{basitext}
3474 { encoding = {T1,LY1,OT1,OT4,T5},
3475   family = {rm*,sf*},
3476   series = {m},
3477   size = {normalsize,footnotesize,small,large}
3478 }
3479
3480 \DeclareMicrotypeSet{normalfont}
3481 { font = */*/*/*/* }
3482
```

The default sets.

```
3483 %%% -----
3484 %%% DEFAULT SETS
3485
3486 \DeclareMicrotypeSetDefault[protrusion]{alltext}
3487 \DeclareMicrotypeSetDefault[expansion]{basitext}
3488 \begin{fontconfig}
3489 \DeclareMicrotypeSetDefault[spacing]{basitext}
3490 \DeclareMicrotypeSetDefault[kerning]{alltext}
3491 \end{fontconfig}
3492
```

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. The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively, also the qfonts package which provides both, and mathdesign's Charter should be the same as Bitstream Charter.

```

3493 %%% -----
3494 %%% FONT ALIASES
3495
3496 \DeclareMicrotypeAlias{lmr}{cmr}% lmodern
3497 \DeclareMicrotypeAlias{aer}{cmr}% ae
3498 \DeclareMicrotypeAlias{zer}{cmr}% zefonts
3499 \DeclareMicrotypeAlias{cmor}{cmr}% eco
3500 \DeclareMicrotypeAlias{hfor}{cmr}% hfoldsty
3501 \DeclareMicrotypeAlias{pxr}{ppl}% pxfonts
3502 \DeclareMicrotypeAlias{qpl}{ppl}% qfonts/QuasiPalatino
3503 \DeclareMicrotypeAlias{txr}{ptm}% txfonts
3504 \DeclareMicrotypeAlias{qtm}{ptm}% qfonts/QuasiTimes
3505 \DeclareMicrotypeAlias{mdbch}{bch}% MathDesign/Charter
3506

```

More Times variants, to be checked: pns, mns (TimesNewRomanPS); mnt (TimesNewRomanMT), mntx (TimesNRExpertMT); mtm (TimesSmallTextMT); pte (Times-Europa); ptt, pttj (TimesTen); TimesEighteen; TimesModernEF.

```

3507 <*/beta>

```

## 14.2 Interaction with babel

```

3508 %%% -----
3509 %%% INTERACTION WITH THE 'babel' PACKAGE
3510
3511 \DeclareMicrotypeBabelHook
3512   {french,français}
3513   {kerning=french, spacing=}
3514
3515 \DeclareMicrotypeBabelHook
3516   {english,american,USenglish,british,UKenglish}
3517   {kerning=, spacing=nonfrench}
3518
3519 \DeclareMicrotypeBabelHook
3520   {turkish}
3521   {kerning=turkish, spacing=}
3522
3523 </beta>

```

## 14.3 Note on Admissible Characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```

\ : \textbackslash
{ : \textbraceleft
} : \textbraceright
^ : \textasciicircum
% : \%
# : \#

```

Comma and equal sign must be guarded with braces (`{,}`, `{=}`) to keep keyval happy.

Character commands are allowed as far as they have been defined in the proper  $\text{\LaTeX}$  way, that is, when they have been assigned a slot in the font encoding with `\DeclareTextSymbol` or `\DeclareTextComposite`. Characters defined via `\chardef` are also possible.

Ligatures and `\mathchardef` symbols have to be specified numerically. Of course, numerical identification is possible in any other case, too.

8-bit characters are also admissible, provided they have been declared in the input encoding file. They should, however, only be used in private configuration files, where the proper input encoding is guaranteed.

## 14.4 Character Inheritance

First the lists of inheriting characters. We only declare those characters that are the same on *both* sides, i. e., not  $\text{\O}$  for  $\text{\O}$ .

```

3524 %%% -----
3525 %%% CHARACTER INHERITANCE
3526

```

### 14.4.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 (`'fi` ligature), 013 (`'fl`), 014 (`'ffi`), 015 (`'ffl`),  $\text{\AE}$ ,  $\text{\ae}$ ,  $\text{\OE}$ ,  $\text{\oe}$ .

```

3527 \DeclareCharacterInheritance
3528   { encoding = OT1 }
3529   { f = {011}, % ff
3530     i = {\i},
3531     j = {\j},
3532     O = {\O},
3533     o = {\o},
3534   }
3535

```

### 14.4.2 T1

Candidates here: 028 (`'fi`), 029 (`'fl`), 030 (`'ffi`), 031 (`'ffl`), 156 (`'IJ` ligature), 188 (`'ij`),  $\text{\AE}$ ,  $\text{\ae}$ ,  $\text{\OE}$ ,  $\text{\oe}$ .

```

3536 \DeclareCharacterInheritance
3537   { encoding = T1 }
3538   { A = {\A,\A,\^A,\~A,\"A,\r{A},\k{A},\u{A}},
3539     a = {\a,\a,\^a,\~a,\"a,\r{a},\k{a},\u{a}},
3540     C = {\C,\c{C},\v{C}},

```

```

3541 c = {\c,\c{c},\v{c}},
3542 D = {\v{D},\DH},
3543 d = {\v{d},\dj},
3544 E = {\'E,\'E,\^E,\"E,\k{E},\v{E}},
3545 e = {\'e,\'e,\^e,\"e,\k{e},\v{e}},
3546 f = {027}, % ff
3547 G = {\u{G}},
3548 g = {\u{g}},
3549 I = {\'I,\'I,\^I,\"I,\.I},
3550 i = {\'i,\'i,\^i,\"i,\.i},
3551 j = {\j},
3552 L = {\L,\'L,\v{L}},
3553 l = {\l,\'l,\v{l}},
3554 N = {\'N,\~N,\v{N}},
3555 n = {\'n,\~n,\v{n}},
3556 O = {\0,\'0,\'0,\^0,\~0,\"0,\H{0}},
3557 o = {\o,\'o,\'o,\^o,\~o,\"o,\H{o}},
3558 R = {\'R,\v{R}},
3559 r = {\'r,\v{r}},
3560 S = {\'S,\c{S},\v{S},\SS},
3561 s = {\'s,\c{s},\v{s}},
3562 T = {\c{T},\v{T}},
3563 t = {\c{t},\v{t}},
3564 U = {\'U,\'U,\^U,\"U,\H{U},\r{U}},
3565 u = {\'u,\'u,\^u,\"u,\H{u},\r{u}},
3566 Y = {\'Y,\"Y},
3567 y = {\'y,\"y},
3568 Z = {\'Z,\.Z,\v{Z}},
3569 z = {\'z,\.z,\v{z}},
3570 - = {127},
3571 }
3572

```

### 14.4.3 LY1

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

```

3573 \DeclareCharacterInheritance
3574 { encoding = LY1 }
3575 { A = {\'A,\'A,\^A,\~A,\"A,\r{A}},
3576 a = {\'a,\'a,\^a,\~a,\"a,\r{a}},
3577 C = {\c{C}},
3578 c = {\c{c}},
3579 D = {\DH},
3580 E = {\'E,\'E,\^E,\"E},
3581 e = {\'e,\'e,\^e,\"e},
3582 f = {011}, % ff
3583 I = {\'I,\'I,\^I,\"I},
3584 i = {\'i,\'i,\^i,\"i,\.i},
3585 L = {\L},
3586 l = {\l},
3587 N = {\~N},
3588 n = {\~n},
3589 O = {\'O,\'O,\^O,\~O,\"O,\O},
3590 o = {\'o,\'o,\^o,\~o,\"o,\o},
3591 S = {\v{S}},
3592 s = {\v{s}},
3593 U = {\'U,\'U,\^U,\"U},
3594 u = {\'u,\'u,\^u,\"u},
3595 Y = {\'Y,\"Y},

```

```

3596     y = {\'y,\^y},
3597     Z = {\v{Z}},
3598     z = {\v{z}},
3599 }
3600

```

#### 14.4.4 OT4

The Polish OT1 extension. More interesting characters here: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

3601 \DeclareCharacterInheritance
3602 { encoding = OT4 }
3603 { A = {\k{A}},
3604   a = {\k{a}},
3605   C = {\'C},
3606   c = {\'c},
3607   E = {\k{E}},
3608   e = {\k{e}},
3609   f = {011}, % ff
3610   i = {\i},
3611   j = {\j},
3612   L = {\L},
3613   l = {\l},
3614   N = {\'N},
3615   n = {\'n},
3616   O = {\0,\'0},
3617   o = {\0,\'o},
3618   S = {\'S},
3619   s = {\'s},
3620   Z = {\'Z,\.Z},
3621   z = {\'z,\.z},
3622 }
3623

```

#### 14.4.5 T5

The Vietnamese encoding T5. It is so crowded with accented and double-accented characters that there is no room for any ligatures.

```

3624 \DeclareCharacterInheritance
3625 { encoding = T5 }
3626 { A = {\'A,\'A,\~A,\h{A},\d{A},\^A,\u{A},
3627       \'\Acircumflex,\'\Acircumflex,\~\Acircumflex,\h\Acircumflex,\d\Acircumflex,
3628       \'\Abreve,\'\Abreve,\~\Abreve,\h\Abreve,\d\Abreve},
3629   a = {\'a,\'a,\~a,\h{a},\d{a},\^a,\u{a},
3630       \'\acircumflex,\'\acircumflex,\~\acircumflex,\h\acircumflex,\d\acircumflex,
3631       \'\abreve,\'\abreve,\~\abreve,\h\abreve,\d\abreve},
3632   D = {\DJ},
3633   d = {\dj},
3634   E = {\'E,\'E,\~E,\h{E},\d{E},\^E,
3635       \'\Ecircumflex,\'\Ecircumflex,\~\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
3636   e = {\'e,\'e,\~e,\h{e},\d{e},\^e,
3637       \'\ecircumflex,\'\ecircumflex,\~\ecircumflex,\h\ecircumflex,\d\ecircumflex},
3638   I = {\'I,\'I,\~I,\h{I},\d{I}},
3639   i = {\'i,\'i,\~i,\h{i},\d{i},\i},
3640   O = {\'O,\'O,\~O,\h{O},\d{O},\^O,\horn{O},
3641       \'\Ocircumflex,\'\Ocircumflex,\~\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
3642       \'\Ohorn,\'\Ohorn,\~\Ohorn,\h\Ohorn,\d\Ohorn},

```

```

3643 o = {\'o,\'o,\~o,\h{o},\d{o},\^o,\horn{o},
3644         \'\ocircumflex,\'\ocircumflex,\~\ocircumflex,\h\ocircumflex,\d\ocircumflex,
3645         \'\ohorn,\'\ohorn,\~\ohorn,\h\ohorn,\d\ohorn},
3646 U = {\'U,\'U,\~U,\h{U},\d{U},\horn{U},
3647         \'\Uhorn,\'\Uhorn,\~\Uhorn,\h\Uhorn,\d\Uhorn},
3648 u = {\'u,\'u,\~u,\h{u},\d{u},\horn{u},
3649         \'\uhorn,\'\uhorn,\~\uhorn,\h\uhorn,\d\uhorn},
3650 Y = {\'Y,\'Y,\~Y,\h{Y},\d{Y}},
3651 y = {\'y,\'y,\~y,\h{y},\d{y}},
3652 }
3653

```

## 14.5 Font Expansion

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

```

3654 %%% -----
3655 %%% EXPANSION SETTINGS
3656
3657 \SetExpansion
3658 <m-t> [ name      = default      ]
3659 <bch> [ name      = bch-default  ]
3660 <cmr> [ name      = cmr-default  ]
3661 <pad> [ name      = pad-default  ]
3662 <pmn> [ name      = pmn-default  ]
3663 <ppl> [ name      = ppl-default  ]
3664 <ptm> [ name      = ptm-default  ]
3665 <m-t> { encoding = {OT1,OT4,T1,LY1} }
3666 <m-t> { encoding = {OT1,T1,LY1},
3667 <bch>   family   = bch }
3668 <cmr>   family   = cmr }
3669 <pad>   family   = {pad,padx,padj} }
3670 <pmn>   family   = {pmn,pmnx,pmnj} }
3671 <ppl>   family   = {ppl,pplx,pplj} }
3672 <ptm>   family   = {ptm,ptmx,ptmj} }
3673 {
3674     A = 500,      a = 700,
3675     \AE = 500,    \ae = 700,
3676     B = 700,      b = 700,
3677     C = 700,      c = 700,
3678     D = 500,      d = 700,
3679     E = 700,      e = 700,
3680     F = 700,
3681     G = 500,      g = 700,
3682     H = 700,      h = 700,
3683     K = 700,      k = 700,
3684     M = 700,      m = 700,
3685     N = 700,      n = 700,
3686     O = 500,      o = 700,
3687     \OE = 500,    \oe = 700,
3688     P = 700,      p = 700,
3689     Q = 500,      q = 700,
3690     R = 700,
3691     S = 700,      s = 700,
3692     U = 700,      u = 700,
3693     W = 700,      w = 700,
3694     Z = 700,      z = 700,
3695     2 = 700,

```

```

3696     3 = 700,
3697     6 = 700,
3698     8 = 700,
3699     9 = 700,
3700   }
3701

```

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

```

3702 \SetExpansion
3703   [ name      = T5 ]
3704   { encoding = T5 }
3705   {
3706     A = 500,      a = 700,
3707     B = 700,      b = 700,
3708     C = 700,      c = 700,
3709     D = 500,      d = 700,
3710     E = 700,      e = 700,
3711     F = 700,
3712     G = 500,      g = 700,
3713     H = 700,      h = 700,
3714     K = 700,      k = 700,
3715     M = 700,      m = 700,
3716     N = 700,      n = 700,
3717     O = 500,      o = 700,
3718     P = 700,      p = 700,
3719     Q = 500,      q = 700,
3720     R = 700,
3721     S = 700,      s = 700,
3722     U = 700,      u = 700,
3723     W = 700,      w = 700,
3724     Z = 700,      z = 700,
3725     2 = 700,
3726     3 = 700,
3727     6 = 700,
3728     8 = 700,
3729     9 = 700,
3730   }
3731
3732 /m-t

```

## 14.6 Character Protrusion

```

3733 %%% -----
3734 %%% PROTRUSION SETTINGS
3735

```

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},
}

```

We also create configuration files for the fonts Bitstream Charter (NFSS code bch), Computer Modern Roman (cmr), Palatino (ppl, pplx, pplj), Times (ptm, ptmx, ptmj), Adobe Garamond (pad, padx, padj) and Minion<sup>16</sup> (pmnx, pmnj), and for the AMS math fonts (msa, msb, euf, eus).

#### 14.6.1 Default

The default settings always use the most moderate value.

```

3736 <!*cfg-u>
3737 \SetProtrusion
3738 <m-t> [ name = default ]
3739 <bch> [ name = bch-default ]
3740 <cmr> [ name = cmr-default ]
3741 <pad> [ name = pad-default ]
3742 <pmn> [ name = pmnj-default ]
3743 <ppl> [ name = ppl-default ]
3744 <ptm> [ name = ptm-default ]
3745 <m-t> { encoding = OT1 }
3746 <cmr> { }
3747 <bch|pad|pmn> { encoding = OT1,
3748 <ppl|ptm> { encoding = {OT1,OT4},
3749 <bch> family = bch }
3750 <pad> family = {pad,padx,padj} }
3751 <pmn> family = pmnj }
3752 <ppl> family = {ppl,pplx,pplj} }
3753 <ptm> family = {ptm,ptmx,ptmj} }
3754 {
3755 A = {50,50},
3756 <m-t|pad|ptm> \AE = {50, },
3757 <bch|pad|pmn> C = {50, },
3758 <bch|pad|pmn> D = { ,50},
3759 <m-t|bch|cmr|pad|pmn|ptm> F = { ,50},
3760 <bch|pad|pmn> G = {50, },
3761 <m-t|cmr|pad|pmn|ppl|ptm> J = {50, },
3762 <bch> J = {100, },
3763 K = { ,50},
3764 <m-t|bch|cmr|pad|pmn|ppl> L = { ,50},

```

<sup>16</sup> Contributed by Harald Harders ([h.harders@tu-bs.de](mailto:h.harders@tu-bs.de))

```

3765 <ptm>      L = { ,80},
3766 <bch|pad|pmn> 0 = {50,50},
3767 <pad|pmn>    \OE = {50, },
3768 <bch|pad|pmn> Q = {50,70},
3769 <bch>       R = { ,50},
3770          T = {50,50},
3771          V = {50,50},
3772          W = {50,50},
3773          X = {50,50},
3774 <m-t|bch|cmr|pad|pmn|ppl> Y = {50,50},
3775 <ptm>       Y = {80,80},
3776          k = { ,50},
3777 <pmn>       l = { , -50},
3778 <pad|ppl>   p = {50,50},
3779 <pad|ppl>   q = {50, },
3780          r = { ,50},
3781 <cmr|pad|pmn> t = { ,70},
3782 <bch>       t = { ,50},
3783          v = {50,50},
3784          w = {50,50},
3785          x = {50,50},
3786 <m-t|bch|pad|pmn> y = { ,50},
3787 <cmr|ppl|ptm> y = {50,70},

3788 <cmr>       0 = { ,50},
3789 <m-t>       1 = {50,50},
3790 <bch|pad|ptm> 1 = {150,150},
3791 <cmr>       1 = {100,200},
3792 <pmn>       1 = { ,50},
3793 <ppl>       1 = {100,100},
3794 <bch|cmr|pad> 2 = {50,50},
3795 <cmr|pad>    3 = {50,50},
3796 <bch|pmn>    3 = {50, },
3797 <m-t|pad>    4 = {50,50},
3798 <bch>       4 = {100,50},
3799 <cmr>       4 = {70,70},
3800 <pmn>       4 = {50, },
3801 <ptm>       4 = {70, },
3802 <cmr>       5 = { ,50},
3803 <pad>       5 = {50,50},
3804 <bch>       6 = {50, },
3805 <cmr>       6 = { ,50},
3806 <pad>       6 = {50,50},
3807 <m-t>       7 = {50,50},
3808 <bch|pad|pmn> 7 = {50,80},
3809 <cmr|ptm>    7 = {50,100},
3810 <ppl>       7 = { ,50},
3811 <cmr>       8 = { ,50},
3812 <bch|pad>    9 = {50,50},
3813 <cmr>       9 = { ,50},
3814 <m-t|cmr|pad|pmn|ppl|ptm> . = { ,700},
3815 <bch>       . = { ,600},
3816          {,} = { ,500},
3817 <m-t|cmr|pad|pmn|ppl|ptm> : = { ,500},
3818 <bch>       : = { ,400},
3819 <m-t|bch|pad|pmn|ptm> ; = { ,300},
3820 <cmr|ppl>    ; = { ,500},
3821          ! = { ,100},
3822 <m-t|pad|pmn|ptm> ? = { ,100},
3823 <bch|cmr|ppl> ? = { ,200},

```

```

3824 <pmn> " = {300,300},
3825 <m-t|bch|cmr|pad|pmn|ppl> @ = {50,50},
3826 <ptm> @ = {100,100},
3827 ~ = {200,250},
3828 <pad|ppl|ptm> & = {50,100},
3829 <m-t|cmr|pad|pmn> \% = {50,50},
3830 <bch> \% = { ,50},
3831 <ppl|ptm> \% = {100,100},
3832 <m-t|ppl|ptm> * = {200,200},
3833 <bch|pmn> * = {200,300},
3834 <cmr|pad> * = {300,300},
3835 <m-t|cmr|ppl|ptm> + = {250,250},
3836 <bch> + = {150,250},
3837 <pad> + = {300,300},
3838 <pmn> + = {150,200},
3839 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
3840 <bch> ( = {200, }, ) = { ,200},
3841 <cmr|ppl> ( = {100, }, ) = { ,300},
3842 <bch|pmn> [ = {100, }, ] = { ,100},

3843 <m-t|pad|pmn|ptm> / = {100,200},
3844 <bch> / = { ,200},
3845 <cmr|ppl> / = {200,300},
3846 <m-t|ptm> - = {500,500},
3847 <bch|cmr|ppl> - = {400,500},
3848 <pad> - = {300,500},
3849 <pmn> - = {200,400},
3850 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
3851 <bch> \textendash = {200,300}, \textendash = {150,250},
3852 <cmr> \textendash = {400,300}, \textendash = {300,200},
3853 <pad|ppl|ptm> \textendash = {300,300}, \textendash = {200,200},

```

Why settings for left *and* right quotes? Because in some languages they might be used like that (see the csquotes package for examples).

```

3854 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
3855 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
3856 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
3857 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
3858 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300},
3859 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600},
3860 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400},
3861 }
3862

```

Greek uppercase letters are in OT1 encoding only.

```

3863 <*cmr>
3864 \SetProtrusion
3865 [ name = cmr-OT1,
3866 load = cmr-default ]
3867 { encoding = {OT1,OT4},
3868 family = cmr }
3869 {
3870 \AE = { 50, },
3871 "00 = { ,150}, % \Gamma
3872 "01 = {100,100}, % \Delta
3873 "02 = { 50, 50}, % \Theta
3874 "03 = {100,100}, % \Lambda
3875 % "04 = { , }, % \Xi
3876 % "05 = { , }, % \Pi
3877 "06 = { 50, 50}, % \Sigma

```

```

3878 "07 = {100,100}, % \Upsilon
3879 "08 = { 50, 50}, % \Phi
3880 "09 = { 50, 50}, % \Psi
3881 % "0A = { , }, % \Omega
3882 }
3883
3884 </cmr>

```

T1 and LY1 encodings contain some more characters. The default list will be loaded first.

```

3885 \SetProtrusion
3886 <m-t> [ name = T1-default,
3887 <bch> [ name = bch-T1,
3888 <cmr> [ name = cmr-T1,
3889 <pad> [ name = pad-T1,
3890 <pmn> [ name = pmnj-T1,
3891 <ppl> [ name = ppl-T1,
3892 <ptm> [ name = ptm-T1,
3893 <m-t> load = default ]
3894 <bch> load = bch-default ]
3895 <cmr> load = cmr-default ]
3896 <pad> load = pad-default ]
3897 <pmn> load = pmnj-default ]
3898 <ppl> load = ppl-default ]
3899 <ptm> load = ptm-default ]
3900 <m-t> { encoding = {T1,LY1} }
3901 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
3902 <ptm> { encoding = {T1},
3903 <bch> family = bch }
3904 <cmr> family = cmr }
3905 <pad> family = {pad,padx,padj} }
3906 <pmn> family = pmnj }
3907 <ppl> family = {ppl,pplx,pplj} }
3908 <ptm> family = {ptm,ptmx,ptmj} }
3909 {
3910 <cmr> \AE = {50, },
3911 <bch> \OE = {50, },
3912 <pmn> \TH = { ,50},
3913 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
3914 <cmr> _ = {200,200},
3915 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
3916 <bch> \textbackslash = {150,200},
3917 <cmr|ppl> \textbackslash = {200,300},
3918 <cmr> \textquotedblleft = {200,600},
3919 <cmr> \textquotedbl = {300,300},

```

The EC fonts do something weird: They insert an implicit kern between quote and boundary character. Therefore, we must override the settings from OT1.

```

3920 <m-t|cmr|pad|ppl|ptm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3921 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
3922 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
3923 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
3924 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
3925 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
3926 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
3927 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
3928 <ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
3929 <m-t|bch|cmr|pad|pmn|ppl> \textexclamdown = {100, }, \textquestiondown = {100, },
3930 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },

```

```

3931 <m-t|cmr|pad|ppl|ptm> \textbraceleft = {400,200}, \textbraceright = {200,400},
3932 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
3933 <m-t|bch|cmr|pad|ppl|ptm> \textless = {200,100}, \textgreater = {100,200},
3934 <pmn> \textless = {100, }, \textgreater = { ,100},
3935 <pmn> \textvisiblespace = {100,100}, % not in LY1
3936 % \dh = { , },
3937 % \th = { , },
3938 % \NG = { , },
3939 % \ng = { , },
3940 % \textasciicircum = { , },
3941 % \textbar = { , },
3942 % \textsterling = { , }, % also in TS1
3943 % \textsection = { , }, % also in TS1
3944 }
3945
3946 <*cmr|bch>

```

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.

```

3947 \SetProtrusion
3948 <cmr> [ name = cmr-T5,
3949 <cmr> load = cmr-default ]
3950 <bch> [ name = bch-T5,
3951 <bch> load = bch-default ]
3952 { encoding = T5,
3953 <cmr> family = cmr }
3954 <bch> family = bch }
3955 {
3956 <bch> _ = {100,100},
3957 <bch> \textbackslash = {150,200},
3958 <cmr> \textbackslash = {200,300},
3959 <cmr> \textquotedblleft = {200,600},
3960 <cmr> \textquotedbl = {300,300},
3961 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
3962 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3963 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
3964 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
3965 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},
3966 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
3967 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
3968 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
3969 \textless = {200,100}, \textgreater = {100,200},
3970 }
3971
3972 </cmr|bch>

```

The lmodern fonts, on the other hand, restore the original kerning from the OT1 fonts, and so do we. Silly, isn't it?

```

3973 <*cmr>
3974 \SetProtrusion
3975 [ name = lmr-T1,
3976 load = cmr-T1 ]
3977 { encoding = {T1,LY1},
3978 family = lmr }
3979 {
3980 \textquotedblleft = {500,300},
3981 \quotedblbase = {500,300},
3982 }
3983

```

```

3984 </cmr>
3985 <*pmn>
3986 \SetProtrusion
3987 [ name      = pmnx-OT1,
3988   load      = pmnj-default ]
3989 { encoding = OT1,
3990   family   = pmnx }
3991 {
3992   1 = {230,180},
3993 }
3994
3995 \SetProtrusion
3996 [ name      = pmnx-T1,
3997   load      = pmnj-T1 ]
3998 { encoding = {T1,LY1},
3999   family   = pmnx }
4000 {
4001   1 = {230,180},
4002 }
4003
4004 </pmn>

```

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

```

4005 <*ptm>
4006 \SetProtrusion
4007 [ name      = ptm-LY1,
4008   load      = ptm-T1 ]
4009 { encoding = LY1,
4010   family   = {ptm,ptmx,ptmj} }
4011 {
4012   - = {100,100},
4013   \texttrademark = {100,100},
4014   \textregistered = {100,100},
4015   \textcopyright = {100,100},
4016   \textdegree = {300,300},
4017   \textminus = {200,200},
4018   \textellipsis = {100,100},
4019   \texteuro = { , }, % ?
4020   \textcent = {100,100},
4021   \textquotesingle = {500,500},
4022   \textflorin = { 50, 70},
4023   \textdagger = {150,150},
4024   \textdaggerdbl = {100,100},
4025   \textperthousand = { , 50},
4026   \textbullet = {150,150},
4027   \textonesuperior = {100,100},
4028   \texttwosuperior = { 50, 50},
4029   \textthreesuperior = { 50, 50},
4030   \textperiodcentered = {300,300},
4031   \textplusminus = { 50, 80},
4032   \textmultiply = {100,100},
4033   \textdivide = { 50,150},
4034 % \textbrokenbar = { , },
4035 % \textyen = { , },
4036 % \textfractionsolidus = { , },
4037 % \textordfeminine = { , },
4038 % \textordmasculine = { , },
4039 % \textmu = { , },
4040 % \textparagraph = { , },

```

```

4041 % \textonequarter      = { , },
4042 % \textonehalf         = { , },
4043 % \textthreequarters   = { , },
4044 }
4045
4046 </ptm>

```

### 14.6.2 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.

```

4047 \SetProtrusion
4048 <m-t> [ name = OT1-it ]
4049 <bch> [ name = bch-it ]
4050 <cmr> [ name = cmr-it ]
4051 <pad> [ name = pad-it ]
4052 <pmn> [ name = pmnj-it ]
4053 <ppl> [ name = ppl-it ]
4054 <ptm> [ name = ptm-it ]
4055 <m-t|bch|pad|pmn> { encoding = OT1,
4056 <cmr> { }
4057 <ppl|ptm> { encoding = {OT1,OT4},
4058 <bch> family = bch,
4059 <pad> family = {pad,padx,padj},
4060 <pmn> family = pmnj,
4061 <ppl> family = {ppl,pplx,pplj},
4062 <ptm> family = {ptm,ptmx,ptmj},
4063 <!cmr> shape = {it,sl} }
4064 {
4065 <cmr|ptm> A = {100,50},
4066 <pad|pmn> A = {50, },
4067 <ppl> A = {50,50},
4068 <ptm> \AE = {100, },
4069 <pad|ppl> \AE = {50, },
4070 <pmn> \AE = { , -50},
4071 <cmr|pad|ppl|ptm> B = {50, },
4072 <pmn> B = {20,-50},
4073 <bch|ppl|ptm> C = {50, },
4074 <cmr|pad> C = {100, },
4075 <pmn> C = {50,-50},
4076 <cmr|pad|ppl|ptm> D = {50,50},
4077 <pmn> D = {20, },
4078 <cmr|pad|ppl|ptm> E = {50, },
4079 <pmn> E = {20,-50},
4080 <cmr|pad|ptm> F = {100, },
4081 <pmn> F = {10, },
4082 <ppl> F = {50, },
4083 <bch|ppl|ptm> G = {50, },
4084 <cmr|pad> G = {100, },
4085 <pmn> G = {50,-50},
4086 <cmr|pad|ppl|ptm> H = {50, },
4087 <cmr|pad|ptm> I = {50, },
4088 <pmn> I = {20,-50},
4089 <cmr|ptm> J = {100, },
4090 <pad> J = {50, },
4091 <pmn> J = {20, },
4092 <cmr|pad|ppl|ptm> K = {50, },

```

```

4093 <pmn>      K = {20,  },
4094 <cmr|pad|ppl|ptm>  L = {50,  },
4095 <pmn>      L = {20,50},
4096 <cmr|ptm>   M = {50,  },
4097 <pmn>      M = {  ,-30},
4098 <cmr|ptm>   N = {50,  },
4099 <pmn>      N = {  ,-30},
4100 <bch|pmn|ppl|ptm>  O = {50,  },
4101 <cmr|pad>   O = {100, },
4102 <pmn|ppl|ptm>  \OE = {50,  },
4103 <pad>      \OE = {100, },
4104 <cmr|pad|ppl|ptm> P = {50,  },
4105 <pmn>      P = {20,-50},
4106 <bch|pmn|ppl|ptm> Q = {50,  },
4107 <cmr|pad>   Q = {100, },
4108 <cmr|pad|ppl|ptm> R = {50,  },
4109 <pmn>      R = {20,  },
4110 <bch|cmr|pad|ppl|ptm> S = {50,  },
4111 <pmn>      S = {20,-30},
4112 <bch|cmr|pad|ppl|ptm> $ = {50,  },
4113 <pmn>      $ = {20,-30},
4114 <bch|pmn>   T = {70,  },
4115 <cmr|pad|ppl|ptm> T = {100, },
4116 <cmr|pad|ppl|ptm> U = {50,  },
4117 <pmn>      U = {50,-50},
4118 <cmr|pad|pmn> V = {100, },
4119 <ppl|ptm>   V = {100,50},
4120 <cmr|pad|pmn> W = {100, },
4121 <ppl>      W = {50,  },
4122 <ptm>      W = {100,50},
4123 <cmr|ppl|ptm> X = {50,  },
4124 <cmr|ptm> Y = {100, },
4125 <pmn>      Y = {50,  },
4126 <ppl>      Y = {100,50},
4127 <pmn>      Z = {  ,-50},
4128 <pmn>      d = {  ,-50},
4129 <pad|pmn>   f = {  ,-100},
4130 <pmn>      i = {  ,-30},
4131 <pmn>      j = {  ,-30},
4132 <pmn>      l = {  ,-100},
4133 <bch>      o = {50,50},
4134 <bch>      p = {  ,50},
4135 <pmn>      p = {-50,  },
4136 <bch>      q = {50,  },
4137 <pmn>      r = {  ,50},
4138 <bch>      t = {  ,50},
4139 <pmn>      v = {50,  },
4140 <bch>      w = {  ,50},
4141 <pmn>      w = {50,  },
4142 <bch>      y = {  ,50},
4143 <cmr>      0 = {100, },
4144 <bch|ptm>   1 = {150,100},
4145 <cmr>      1 = {200,50},
4146 <pad>      1 = {150,  },
4147 <pmn>      1 = {50,  },
4148 <ppl>      1 = {100, },
4149 <cmr>      2 = {100,-100},
4150 <pad|ppl|ptm> 2 = {50,  },
4151 <pmn>      2 = {-50,  },
4152 <bch>      3 = {50,  },

```



```

4153 <cmr>      3 = {100,-100},
4154 <pmn>      3 = {-100, },
4155 <ptm>      3 = {100,50},
4156 <bch>      4 = {100, },
4157 <cmr|pad>   4 = {150, },
4158 <ppl|ptm>   4 = {50, },
4159 <cmr>      5 = {100, },
4160 <ptm>      5 = {50, },
4161 <bch>      6 = {50, },
4162 <cmr>      6 = {100, },
4163 <bch|pad|ptm> 7 = {100, },
4164 <cmr>      7 = {200,-150},
4165 <pmn>      7 = {20, },
4166 <ppl>      7 = {50, },
4167 <cmr>      8 = {50,-50},
4168 <cmr>      9 = {100,-100},
4169 <m-t|cmr|pad|pmn|ppl> . = { ,500},
4170 <bch|ptm> . = { ,700},
4171 <m-t|cmr|pad|pmn|ppl> {,}= { ,500},
4172 <bch> {,}= { ,600},
4173 <ptm> {,}= { ,700},
4174 <m-t|cmr|pad|ppl> : = { ,300},
4175 <bch> : = { ,400},
4176 <pmn> : = { ,200},
4177 <ptm> : = { ,500},
4178 <m-t|cmr|pad|ppl> ; = { ,300},
4179 <bch> ; = { ,400},
4180 <pmn> ; = { ,200},
4181 <ptm> ; = { ,500},
4182 <ptm> ! = { ,100},
4183 <bch> ? = { ,200},
4184 <ptm> ? = { ,100},
4185 <ppl> ? = { ,300},
4186 <pmn> " = {400,200},
4187 <m-t|pad|pmn|ppl|ptm> & = {50,50},
4188 <bch> & = { ,80},
4189 <cmr> & = {100,50},
4190 <m-t|cmr|pad|pmn> \% = {100, },
4191 <bch> \% = {50,50},
4192 <ppl|ptm> \% = {100,100},
4193 <m-t|pmn|ppl> * = {200,200},
4194 <bch> * = {300,200},
4195 <cmr> * = {400,100},
4196 <pad> * = {500,100},
4197 <ptm> * = {400,200},
4198 <m-t|cmr|pmn|ppl> + = {150,200},
4199 <bch> + = {250,250},
4200 <pad|ptm> + = {250,200},
4201 <m-t|pad|pmn|ppl> @ = {50,50},
4202 <bch> @ = {80,50},
4203 <cmr> @ = {200,50},
4204 <ptm> @ = {150,150},
4205 <m-t|bch> ~ = {150,150},
4206 <cmr|pad|pmn|ppl|ptm> - = {200,150},
4207 ( = {200, }, ) = { ,200},
4208 <m-t|cmr|pad|ppl|ptm> / = {100,200},
4209 <bch> / = { ,150},
4210 <pmn> / = {100,150},
4211 <m-t> - = {300,300},
4212 <bch|pad> - = {300,400},

```

```

4213 <pmn> - = {200,300},
4214 <cmr> - = {500,300},
4215 <ppl> - = {300,500},
4216 <ptm> - = {500,500},
4217 <m-t|pmn> \textendash = {200,200}, \textemdash = {150,150},
4218 <bch> \textendash = {200,300}, \textemdash = {150,200},
4219 <cmr> \textendash = {500,300}, \textemdash = {400,200},
4220 <pad|ppl|ptm> \textendash = {300,300}, \textemdash = {200,200},
4221 <m-t|bch|pmn> \textquoteleft = {400,200}, \textquoteright = {400,200},
4222 <cmr|pad> \textquoteleft = {800,200}, \textquoteright = {800,200},
4223 <ppl> \textquoteleft = {700,400}, \textquoteright = {700,400},
4224 <ptm> \textquoteleft = {800,500}, \textquoteright = {800,500},
4225 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200},
4226 <cmr> \textquotedblleft = {700,100}, \textquotedblright = {500,300},
4227 <pad> \textquotedblleft = {700,200}, \textquotedblright = {700,200},
4228 <ppl> \textquotedblleft = {500,300}, \textquotedblright = {500,300},
4229 <ptm> \textquotedblleft = {700,400}, \textquotedblright = {700,400},
4230 }
4231
4232 <*cmr>
4233 \SetProtrusion
4234 [ name = cmr-it-OT1,
4235 load = cmr-it ]
4236 { encoding = {OT1,OT4},
4237 family = cmr,
4238 shape = it }
4239 {
4240 \AE = {100, },
4241 \OE = {100, },
4242 "00 = {200,150}, % \Gamma
4243 "01 = {150,100}, % \Delta
4244 "02 = {150, 50}, % \Theta
4245 "03 = {150, 50}, % \Lambda
4246 "04 = {100,100}, % \Xi
4247 "05 = {100,100}, % \Pi
4248 "06 = {100, 50}, % \Sigma
4249 "07 = {200,150}, % \Upsilon
4250 "08 = {150, 50}, % \Phi
4251 "09 = {150,100}, % \Psi
4252 "0A = { 50, 50}, % \Omega
4253 }
4254
4255 </cmr>
4256 \SetProtrusion
4257 <m-t> [ name = T1-it-default,
4258 <bch> [ name = bch-it-T1,
4259 <cmr> [ name = cmr-it-T1,
4260 <pad> [ name = pad-it-T1,
4261 <pmn> [ name = pmnj-it-T1,
4262 <ppl> [ name = ppl-it-T1,
4263 <ptm> [ name = ptm-it-T1,
4264 <m-t> load = OT1-it ]
4265 <bch> load = bch-it ]
4266 <cmr> load = cmr-it ]
4267 <pmn> load = pmnj-it ]
4268 <pad> load = pad-it ]
4269 <ppl> load = ppl-it ]
4270 <ptm> load = ptm-it ]
4271 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4272 <ptm> { encoding = {T1},

```

```

4273 <bch> family = bch,
4274 <cmr> family = cmr,
4275 <pmn> family = pmnj,
4276 <pad> family = {pad,padx,padj},
4277 <ppl> family = {ppl,pplx,pplj},
4278 <ptm> family = {ptm,ptmx,ptmj},
4279 <!cmr> shape = {it,sl} }
4280 <cmr> shape = it }
4281 {
4282 <m-t|bch|pmn> _ = { ,100},
4283 <cmr> _ = {100,200},
4284 <pad|ppl|ptm> _ = {100,100},
4285 <cmr> \AE = {100, },
4286 <bch> \OE = { 50, },
4287 <cmr> \OE = {100, },
4288 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
4289 <cmr> \textbackslash = {300,300},
4290 <bch> \textbackslash = {150,150},
4291 <pmn> \textbackslash = {100,150},
4292 <pmn> 031 = { , -100}, % ffi
4293 <cmr|ptm> 156 = {100, }, % IJ
4294 <pad> 156 = {50, }, % IJ
4295 <pmn> 156 = {20, }, % IJ
4296 <pmn> 188 = { , -30}, % ij
4297 <pmn> \v{t} = { ,100},
4298 <cmr> \textquotedblleft = {500,300},
4299 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
4300 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4301 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4302 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
4303 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsingright = {300,500},
4304 <bch|pmn> \guilsinglleft = {300,400}, \guilsingright = {200,500},
4305 <cmr> \guilsinglleft = {500,300}, \guilsingright = {400,400},
4306 <pad> \guilsinglleft = {500,400}, \guilsingright = {300,500},
4307 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
4308 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
4309 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4310 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
4311 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
4312 <m-t|pad|ppl> \textexclamdown = {100, }, \textquestiondown = {200, },
4313 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4314 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
4315 <m-t|ppl> \textbraceleft = {200,100}, \textbraceright = {200,200},
4316 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
4317 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
4318 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
4319 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100},
4320 <pmn> \textvisiblespace = {100,100},
4321 }
4322
4323 <*cmr|bch>
4324 \SetProtrusion
4325 <cmr> [ name = cmr-it-T5,
4326 <cmr> load = cmr-it ]
4327 <bch> [ name = bch-it-T5,
4328 <bch> load = bch-it ]
4329 { encoding = T5,
4330 <bch> family = bch,
4331 <cmr> family = cmr,
4332 shape = it }

```

```

4333 {
4334   <bch>      _ = { ,100},
4335   <cmr>      _ = {100,200},
4336   <bch>      \textbackslash = {150,150},
4337   <cmr>      \textbackslash = {300,300},
4338   <bch>      \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4339   <cmr>      \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4340   <bch>      \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4341   <cmr>      \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4342   <bch>      \guillemotleft = {200,300}, \guillemotright = {150,400},
4343   <cmr>      \guillemotleft = {400,100}, \guillemotright = {200,300},
4344   <bch>      \textbraceleft = {200, }, \textbraceright = { ,200},
4345   <cmr>      \textbraceleft = {400,100}, \textbraceright = {200,200},
4346   <bch>      \textless = {100, }, \textgreater = { ,100},
4347   <cmr>      \textless = {300,100}, \textgreater = {200,100},
4348 }
4349
4350
4351 </cmr><bch>

```

Slanted is very similar to italic.

```

4352 <*cmr>
4353 \SetProtrusion
4354 [ name = cmr-sl,
4355   load = cmr-it-OT1 ]
4356 { encoding = {OT1,OT4},
4357   family = cmr,
4358   shape = sl }
4359 {
4360   L = { ,50},
4361   f = { ,-50},
4362   - = {300, },
4363   \textendash = {400, }, \textemdash = {300, },
4364 }
4365
4366 \SetProtrusion
4367 [ name = cmr-sl-T1,
4368   load = cmr-it-T1 ]
4369 { encoding = {T1,LV1},
4370   family = cmr,
4371   shape = sl }
4372 {
4373   L = { ,50},
4374   f = { ,-50},
4375   - = {300, },
4376   \textendash = {400, }, \textemdash = {300, },
4377 }
4378
4379 \SetProtrusion
4380 [ name = cmr-sl-T5,
4381   load = cmr-it-T5 ]
4382 { encoding = T5,
4383   family = cmr,
4384   shape = sl }
4385 {
4386   L = { ,50},
4387   f = { ,-50},
4388   - = {300, },
4389   \textendash = {400, }, \textemdash = {300, },
4390 }

```

```

4391
4392 \SetProtrusion
4393 [ name = lmr-it-T1,
4394   load = cmr-it-T1 ]
4395 { encoding = {T1,LV1},
4396   family = lmr,
4397   shape = {it,sl} }
4398 {
4399   \textquotedblleft = {700,100},
4400   \quotedblbase = {600,300},
4401 }
4402

```

Oldstyle numerals are slightly different.

```

4403 \SetProtrusion
4404 [ name = cmr(oldstyle)-it,
4405   load = cmr-it-T1 ]
4406 { encoding = T1,
4407   family = {hfor,cmor},
4408   shape = {it,sl} }
4409 {
4410   1 = {250, 50},
4411   2 = {150,-100},
4412   3 = {100,-50},
4413   4 = {150,150},
4414   6 = {200,  },
4415   7 = {200, 50},
4416   8 = {150,-50},
4417   9 = {100, 50},
4418 }
4419
4420 /cmr
4421 *pmn
4422 \SetProtrusion
4423 [ name = pmnx-it,
4424   load = pmnj-it ]
4425 { encoding = OT1,
4426   family = pmnx,
4427   shape = {it,sl} }
4428 {
4429   1 = {100,150},
4430 }
4431
4432 \SetProtrusion
4433 [ name = pmnx-it-T1,
4434   load = pmnj-it-T1 ]
4435 { encoding = {T1,LV1},
4436   family = pmnx,
4437   shape = {it,sl} }
4438 {
4439   1 = {100,150},
4440 }
4441
4442 /pmn
4443 *ptm
4444 \SetProtrusion
4445 [ name = ptm-it-LV1,
4446   load = ptm-it-T1 ]
4447 { encoding = {LV1},
4448   family = {ptm,ptmx,ptmj},

```

```

4449     shape      = {it,sl} }
4450     {
4451         _                = {100,100},
4452         \texttrademark    = {100,100},
4453         \textregistered   = {100,100},
4454         \textcopyright     = {100,100},
4455         \textdegree       = {300,100},
4456         \textminus        = {200,200},
4457         \textellipsis     = {100,150},
4458         \texteuro         = { , },
4459         \textcent         = {100,100},
4460         \textquotesingle  = {500, },
4461         \textflorin       = {100, 70},
4462         \textdagger       = {150,150},
4463         \textdaggerdbl    = {100,100},
4464         \textbullet       = {150,150},
4465         \textonesuperior  = {150,100},
4466         \texttwosuperior  = {150, 50},
4467         \textthreesuperior = {150, 50},
4468         \textparagraph    = {100, },
4469         \textperiodcentered = {500,300},
4470         \textonequarter   = { 50, },
4471         \textonehalf      = { 50, },
4472         \textplusminus    = {100,100},
4473         \textmultiply     = {150,150},
4474         \textdivide      = {150,150},
4475     }
4476
4477 </ptm>

```

### 14.6.3 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).

```

4478 \SetProtrusion
4479 <m-t> [ name      = OT1-sc,
4480 <bch> [ name      = bch-sc,
4481 <cmr> [ name      = cmr-sc-OT1,
4482 <pad> [ name      = pad-sc,
4483 <pmn> [ name      = pmnj-sc,
4484 <ppl> [ name      = ppl-sc,
4485 <ptm> [ name      = ptm-sc,
4486 <m-t> load      = default ]
4487 <bch> load      = bch-default ]
4488 <cmr> load      = cmr-OT1 ]
4489 <pad> load      = pad-default ]
4490 <pmn> load      = pmnj-default ]
4491 <ppl> load      = ppl-default ]
4492 <ptm> load      = ptm-default ]
4493 <m-t|bch|pad|pmn> { encoding = OT1,
4494 <cmr|ppl|ptm> { encoding = {OT1,OT4},
4495 <bch> family     = bch,
4496 <cmr> family     = cmr,
4497 <pad> family     = {pad,padx,padj},
4498 <pmn> family     = pmnj,
4499 <ppl> family     = {ppl,pplx,pplj},
4500 <ptm> family     = {ptm,ptmx,ptmj},

```

```

4501     shape      = sc }
4502     {
4503         a = {50,50},
4504         <cmr|pad|ppl|ptm> \ae = {50, },
4505         <bch|pmn> c = {50, },
4506         <bch|pad|pmn> d = { ,50},
4507         <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
4508         <bch|pad|pmn> g = {50, },
4509         <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
4510         <bch> j = {100, },
4511         <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
4512         <ptm> l = { ,80},
4513         <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
4514         <ptm> 013 = { ,80}, % fl
4515         <bch|pad|pmn> o = {50,50},
4516         <pad|pmn> \oe = {50, },
4517         <ppl> p = { 0, 0},
4518         <bch|pad|pmn> q = {50,70},
4519         <ppl> q = { 0, },
4520         <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
4521         t = {50,50},
4522         <m-t|bch|cmr|pad|pmn|ppl> y = {50,50},
4523         <ptm> y = {80,80},
4524     }
4525
4526 \SetProtrusion
4527 <m-t> [ name      = T1-sc,
4528 <bch> [ name      = bch-sc-T1,
4529 <cmr> [ name      = cmr-sc-T1,
4530 <pad> [ name      = pad-sc-T1,
4531 <pmn> [ name      = pmnj-sc-T1,
4532 <ppl> [ name      = ppl-sc-T1,
4533 <ptm> [ name      = ptm-sc-T1,
4534 <m-t> load      = T1-default ]
4535 <bch> load      = bch-T1 ]
4536 <cmr> load      = cmr-T1 ]
4537 <pad> load      = pad-T1 ]
4538 <pmn> load      = pmnj-T1 ]
4539 <ppl> load      = ppl-T1 ]
4540 <ptm> load      = ptm-T1 ]
4541 { encoding = {T1,LY1},
4542 <bch> family  = bch,
4543 <cmr> family  = cmr,
4544 <pad> family  = {pad,padx,padj},
4545 <pmn> family  = pmnj,
4546 <ppl> family  = {ppl,pplx,pplj},
4547 <ptm> family  = {ptm,ptmx,ptmj},
4548     shape      = sc }
4549     {
4550         a = {50,50},
4551         <cmr|pad|ppl|ptm> \ae = {50, },
4552         <bch|pmn> c = {50, },
4553         <bch|pad|pmn> d = { ,50},
4554         <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
4555         <bch|pad|pmn> g = {50, },
4556         <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
4557         <bch> j = {100, },
4558         <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
4559         <ptm> l = { ,80},
4560         <m-t|bch|cmr|pad|pmn|ppl> 029 = { ,50}, % fl

```

```

4561 <ptm> 029 = { ,80}, % f1
4562 <bch|pad|pmn> o = {50,50},
4563 <bch|pad|pmn> \oe = {50, },
4564 <ppl> p = { 0, 0},
4565 <bch|pad|pmn> q = {50,70},
4566 <ppl> q = { 0, },
4567 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
4568 t = {50,50},
4569 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50},
4570 <ptm> y = {80,80},
4571 }
4572
4573 <*cmr|bch>
4574 \SetProtrusion
4575 <bch> [ name = bch-sc-T5,
4576 <bch> load = bch-T5 ]
4577 <cmr> [ name = cmr-sc-T5,
4578 <cmr> load = cmr-T5 ]
4579 { encoding = T5,
4580 <bch> family = bch,
4581 <cmr> family = cmr,
4582 shape = sc }
4583 {
4584 a = {50,50},
4585 <bch> c = {50, },
4586 <bch> d = { ,50},
4587 f = { ,50},
4588 <bch> g = {50, },
4589 <bch> j = {100, },
4590 <cmr> j = {50, },
4591 l = { ,50},
4592 <bch> o = {50,50},
4593 <bch> q = { 0, },
4594 <cmr> r = { , 0},
4595 t = {50,50},
4596 y = {50,50},
4597 }
4598
4599 </cmr|bch>
4600 <*pmn>
4601 \SetProtrusion
4602 [ name = pmnx-sc,
4603 load = pmnj-sc ]
4604 { encoding = OT1,
4605 family = pmnx,
4606 shape = sc }
4607 {
4608 l = {230,180},
4609 }
4610
4611 \SetProtrusion
4612 [ name = pmnx-sc-T1,
4613 load = pmnj-sc-T1 ]
4614 { encoding = {T1,LV1},
4615 family = pmnx,
4616 shape = sc }
4617 {
4618 l = {230,180},
4619 }
4620

```



### 14.6.4 Italic Small Caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's `fontinstallationguide` suggests `si`.

```

4621 \SetProtrusion
4622 [ name      = pmnj-scit,
4623   load      = pmnj-it   ]
4624 { encoding = OT1,
4625   family   = pmnj,
4626   shape     = {scit,si} }
4627 {
4628   a = {50, },
4629   \ae = { , -50},
4630   b = {20, -50},
4631   c = {50, -50},
4632   d = {20, 0},
4633   e = {20, -50},
4634   f = {10, 0},
4635   012 = {10, -50}, % fi
4636   013 = {10, -50}, % fl
4637   014 = {10, -50}, % ffi
4638   015 = {10, -50}, % ffl
4639   g = {50, -50},
4640   i = {20, -50},
4641   j = {20, 0},
4642   k = {20, },
4643   l = {20, 50},
4644   m = { , -30},
4645   n = { , -30},
4646   o = {50, },
4647   \oe = {50, -50},
4648   p = {20, -50},
4649   q = {50, },
4650   r = {20, 0},
4651   s = {20, -30},
4652   t = {70, },
4653   u = {50, -50},
4654   v = {100, },
4655   w = {100, },
4656   y = {50, },
4657   z = { , -50},
4658 }
4659
4660 \SetProtrusion
4661 [ name      = pmnj-scit-T1,
4662   load      = pmnj-it-T1 ]
4663 { encoding = {T1,LY1},
4664   family   = pmnj,
4665   shape     = {scit,si}   }
4666 {
4667   a = {50, },
4668   \ae = { , -50},
4669   b = {20, -50},
4670   c = {50, -50},
4671   d = {20, 0},
4672   e = {20, -50},
4673   f = {10, 0},
4674   028 = {10, -50}, % fi
4675   029 = {10, -50}, % fl

```

```

4676 030 = {10,-50}, % ffi
4677 031 = {10,-50}, % ffi
4678 g = {50,-50},
4679 i = {20,-50},
4680 188 = {20, 0}, % ij
4681 j = {20, 0},
4682 k = {20, },
4683 l = {20,50},
4684 m = { , -30},
4685 n = { , -30},
4686 o = {50, },
4687 \oe = {50,-50},
4688 p = {20,-50},
4689 q = {50, },
4690 r = {20, 0},
4691 s = {20,-30},
4692 t = {70, },
4693 u = {50,-50},
4694 v = {100, },
4695 w = {100, },
4696 y = {50, },
4697 z = { , -50},
4698 }
4699
4700 \SetProtrusion
4701 [ name = pmnx-scit,
4702 load = pmnj-scit ]
4703 { encoding = OT1,
4704 family = pmnx,
4705 shape = {scit,si} }
4706 {
4707 1 = {100,150},
4708 }
4709
4710 \SetProtrusion
4711 [ name = pmnx-scit-T1,
4712 load = pmnj-scit-T1 ]
4713 { encoding = {T1,LV1},
4714 family = pmnx,
4715 shape = {scit,si} }
4716 {
4717 1 = {100,150},
4718 }
4719
4720 </pmn>

```

#### 14.6.5 textcomp

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

4721 \SetProtrusion
4722 <m-t> [ name = textcomp ]
4723 <bch> [ name = bch-textcomp ]
4724 <cmr> [ name = cmr-textcomp ]
4725 <pad> [ name = pad-textcomp ]
4726 <pmn> [ name = pmn-textcomp ]
4727 <ppl> [ name = ppl-textcomp ]
4728 <ptm> [ name = ptm-textcomp ]

```

```

4729 <m-t> { encoding = TS1      }
4730 <!m-t> { encoding = TS1,
4731 <bch>    family   = bch }
4732 <cmr>    family   = cmr }
4733 <pad>    family   = {pad,padx,padj} }
4734 <pmn>    family   = {pmnx,pmnj} }
4735 <ppl>    family   = {ppl,pplx,pplj} }
4736 <ptm>    family   = {ptm,ptmx,ptmj} }
4737 {
4738 <cmr>    \textquotestraightbase = {300,300},
4739 <pad|pmn> \textquotestraightbase = {400,400},
4740 <cmr|pmn> \textquotestraightdblbase = {300,300},
4741 <pad>    \textquotestraightdblbase = {400,400},
4742 <bch|cmr|pad|pmn> \texttwelveudash = {200,200},
4743 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
4744 <cmr|pmn> \textquotesingle = {300,400},
4745 <pad>    \textquotesingle = {400,500},
4746 <ptm>    \textquotesingle = {500,500},
4747 <bch|cmr|pmn> \textasteriskcentered = {200,300},
4748 <pad>    \textasteriskcentered = {300,300},
4749 <pmn>    \textfractionsolidus = {-200,-200},
4750 <cmr>    \textoneoldstyle = {100,100},
4751 <pmn>    \textoneoldstyle = { , 50},
4752 <cmr>    \textthreeoldstyle = { , 50},
4753 <pad|pmn> \textthreeoldstyle = { 50, },
4754 <cmr>    \textfouroldstyle = { 50, 50},
4755 <pad|pmn> \textfouroldstyle = { 50, },
4756 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
4757 <cmr>    \textlangle = {400, },
4758 <cmr>    \textrangle = { , 400},
4759 <m-t|bch|pmn|ptm> \textminus = {200,200},
4760 <cmr|pad|ppl> \textminus = {300,300},
4761 <bch|pad|pmn> \textlbrackdbl = {100, },
4762 <bch|pad|pmn> \textrbrackdbl = { , 100},
4763 <pmn>    \textasciigrave = {200,500},
4764 <bch|cmr|pad|pmn> \texttildebelow = {200,250},
4765 <pmn>    \textasciibreve = {300,400},
4766 <pmn>    \textasciicaron = {300,400},
4767 <pmn>    \textacutedbl = {200,300},
4768 <pmn>    \textgravedbl = {150,300},
4769 <bch|pmn> \textdagger = { 80, 80},
4770 <cmr|pad> \textdagger = {100,100},
4771 <ptm>    \textdagger = {150,150},
4772 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
4773 <ptm>    \textdaggerdbl = {100,100},
4774 <bch>    \textbardbl = {100,100},
4775 <bch>    \textbullet = {200,200},
4776 <cmr|pad|pmn> \textbullet = { , 100},
4777 <ptm>    \textbullet = {150,150},
4778 <bch|cmr|pmn> \textcelsius = { 50, },
4779 <pad>    \textcelsius = { 80, },
4780 <bch>    \textflorin = { 50, 50},
4781 <pad>    \textflorin = { , 100},
4782 <pmn>    \textflorin = { 50, 100},
4783 <ptm>    \textflorin = { 50, 70},
4784 <cmr>    \textcolonmonetary = { , 50},
4785 <pad|pmn> \textcolonmonetary = { 50, },
4786 <pmn>    \textinterrobang = { , 100},
4787 <pmn>    \textinterrobangdown = {100, },
4788 <m-t|pad|ptm> \texttrademark = {100,100},

```

```

4789 <bch> \texttrademark = {150,150},
4790 <cmr|ppl> \texttrademark = {200,200},
4791 <pmn> \texttrademark = { 50, 50},
4792 <bch> \textcent = { 50, },
4793 <ptm> \textcent = {100,100},
4794 <bch> \textsterling = { 50, },
4795 <bch> \textbrokenbar = {200,200},
4796 <pmn> \textasciidieresis = {300,400},
4797 <m-t|bch|cmr|pad|ptm> \textcopyright = {100,100},
4798 <pmn> \textcopyright = {100,150},
4799 <ppl> \textcopyright = {200,200},
4800 <bch|cmr> \textordfeminine = {100,200},
4801 <pad|pmn> \textordfeminine = {200,200},
4802 <bch|cmr|pad|pmn> \textlnot = {200, },
4803 <m-t|bch|cmr|pad|ptm> \textregistered = {100,100},
4804 <pmn> \textregistered = { 50,150},
4805 <ppl> \textregistered = {200,200},
4806 <pmn> \textasciimacron = {150,200},
4807 <m-t|ppl|ptm> \textdegree = {300,300},
4808 <bch> \textdegree = {150,200},
4809 <cmr|pad> \textdegree = {400,400},
4810 <pmn> \textdegree = {150,400},
4811 <bch|cmr|pad|pmn> \textpm = {150,200},
4812 <ptm> \textpm = { 50, 80},
4813 <bch> \texttwosuperior = {100,200},
4814 <cmr> \texttwosuperior = { 50,100},
4815 <pad|pmn> \texttwosuperior = {200,200},
4816 <ptm> \texttwosuperior = { 50, 50},
4817 <bch> \textthreesuperior = {100,200},
4818 <cmr> \textthreesuperior = { 50,100},
4819 <pad|pmn> \textthreesuperior = {200,200},
4820 <ptm> \textthreesuperior = { 50, 50},
4821 <pmn> \textasciicute = {300,400},
4822 <bch> \textmu = { ,100},
4823 <bch|pad|pmn> \textparagraph = { ,100},
4824 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
4825 <ptm> \textperiodcentered = {300,300},
4826 <bch> \textonesuperior = {200,300},
4827 <cmr|pad|pmn> \textonesuperior = {200,200},
4828 <ptm> \textonesuperior = {100,100},
4829 <bch|pad|pmn> \textordmasculine = {200,200},
4830 <cmr> \textordmasculine = {100,200},
4831 <bch|cmr|pmn> \texteuro = {100, },
4832 <pad> \texteuro = { 50,100},
4833 <bch|ptm> \texttimes = {100,100},
4834 <cmr> \texttimes = {150,250},
4835 <pad> \texttimes = {100,150},
4836 <pmn> \texttimes = { 70,100},
4837 <bch|pad|pmn> \textdiv = {150,200},
4838 <cmr> \textdiv = {150,250},
4839 <ptm> \textdiv = { 50,100},
4840 <ptm> \textperthousand = { ,50},

```

All remaining characters can be found in the source.

```

4841 }
4842
4843 <*cmr|pad|pmn>
4844 \SetProtrusion
4845 <cmr> [ name = cmr-textcomp-it ]
4846 <pad> [ name = pad-textcomp-it ]

```

```

4847 <pmn> [ name      = pmn-textcomp-it ]
4848 { encoding = TS1,
4849   family   = cmr,
4850   family   = {pad,padx,padj},
4851   family   = {pmnx,pmnj},
4852   shape     = {it,sl} }
4853 {
4854   <cmr>      \textquotestraightbase = {300,600},
4855   <pad|pmn>  \textquotestraightbase = {400,400},
4856   <cmr>      \textquotestraightdblbase = {300,600},
4857   <pad>      \textquotestraightdblbase = {300,400},
4858   <pmn>      \textquotestraightdblbase = {300,300},
4859   \texttwelvewdash = {200,200},
4860   \textthreequartersemdash = {150,150},
4861   <cmr>      \textquotesingle = {600,300},
4862   <pad>      \textquotesingle = {800,100},
4863   <pmn>      \textquotesingle = {300,200},
4864   <cmr>      \textasteriskcentered = {300,200},
4865   <pad>      \textasteriskcentered = {500,100},
4866   <pmn>      \textasteriskcentered = {200,300},
4867   <pmn>      \textfractionsolidus = {-200,-200},
4868   <cmr>      \textoneoldstyle = {100, 50},
4869   <pad>      \textoneoldstyle = {100,  },
4870   <pmn>      \textoneoldstyle = { 50,  },
4871   <pad>      \texttwooldstyle = { 50,  },
4872   <pmn>      \texttwooldstyle = {-50,  },
4873   <cmr>      \textthreeoldstyle = {100, 50},
4874   <pmn>      \textthreeoldstyle = {-100,  },
4875   <cmr>      \textfouroldstyle = { 50, 50},
4876   <pad>      \textfouroldstyle = { 50,100},
4877   <cmr>      \textsevenoldstyle = { 50, 80},
4878   <pad>      \textsevenoldstyle = { 50,  },
4879   <pmn>      \textsevenoldstyle = { 20,  },
4880   <cmr>      \textlangle = {400,  },
4881   <cmr>      \textrangle = {  ,400},
4882   <cmr|pad>  \textminus = {300,300},
4883   <pmn>      \textminus = {200,200},
4884   <pad|pmn>  \textlbrackdbl = {100,  },
4885   <pad|pmn>  \textrbrackdbl = {  ,100},
4886   <pmn>      \textasciigrave = {300,300},
4887   \texttildelow = {200,250},
4888   <pmn>      \textasciibreve = {300,300},
4889   <pmn>      \textasciicaron = {300,300},
4890   <pmn>      \textacutedbl = {200,300},
4891   <pmn>      \textgravedbl = {150,300},
4892   <cmr>      \textdagger = {100,100},
4893   <pad>      \textdagger = {200,100},
4894   <pmn>      \textdagger = { 80, 50},
4895   <cmr|pad>  \textdaggerdbl = { 80, 80},
4896   <pmn>      \textdaggerdbl = { 80, 50},
4897   <cmr>      \textbullet = {200,100},
4898   <pad>      \textbullet = {300,  },
4899   <pmn>      \textbullet = { 30, 70},
4900   <cmr>      \textcelsius = {100,  },
4901   <pad>      \textcelsius = {200,  },
4902   <pmn>      \textcelsius = { 50,-50},
4903   <pad>      \textflorin = {100,  },
4904   <pmn>      \textflorin = { 50,100},
4905   <cmr>      \textcolonmonetary = {150,  },
4906   <pad>      \textcolonmonetary = {100,  },

```

```

4907 <pmn> \textcolonmonetary = { 50,-50},
4908 <cmr|pad> \texttrademark = {200, },
4909 <pmn> \texttrademark = { 50,100},
4910 <pmn> \textasciidieresis = {300,200},
4911 <cmr> \textcopyright = {100, },
4912 <pad> \textcopyright = {200,100},
4913 <pmn> \textcopyright = {100,150},
4914 <cmr> \textordfeminine = {100,100},
4915 <pmn> \textordfeminine = {200,200},
4916 <cmr|pad> \textlnot = {300, },
4917 <pmn> \textlnot = {200, },
4918 <cmr> \textregistered = {100, },
4919 <pad> \textregistered = {200,100},
4920 <pmn> \textregistered = { 50,150},
4921 <pmn> \textasciimacron = {150,200},
4922 <cmr|pad> \textdegree = {500,100},
4923 <pmn> \textdegree = {150,150},
4924 <cmr> \textpm = {150,100},
4925 <pad> \textpm = {200,150},
4926 <pmn> \textpm = {150,200},
4927 <cmr> \textonesuperior = {400, },
4928 <pad> \textonesuperior = {300,100},
4929 <pmn> \textonesuperior = {200,100},
4930 <cmr> \texttwosuperior = {400, },
4931 <pad> \texttwosuperior = {300, },
4932 <pmn> \texttwosuperior = {200,100},
4933 <cmr> \textthreesuperior = {400, },
4934 <pad> \textthreesuperior = {300, },
4935 <pmn> \textthreesuperior = {200,100},
4936 <pmn> \textasciacute = {300,200},
4937 <cmr> \textparagraph = {200, },
4938 <pmn> \textparagraph = { ,100},
4939 <cmr> \textperiodcentered = {500,500},
4940 <pad|pmn> \textperiodcentered = {300,400},
4941 <cmr> \textordmasculine = {100,100},
4942 <pmn> \textordmasculine = {200,200},
4943 <cmr> \texteuro = {200, },
4944 <pad> \texteuro = {100, },
4945 <pmn> \texteuro = {100,-50},
4946 <cmr> \texttimes = {200,200},
4947 <pad> \texttimes = {200,100},
4948 <pmn> \texttimes = { 70,100},
4949 <cmr|pad> \textdiv = {200,200},
4950 <pmn> \textdiv = {150,200},
4951 }
4952
4953 </cmr|pad|pmn>

```

### 14.6.6 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}
```

```
4954 <*cmr>
4955 \SetProtrusion
4956   [ name      = cmr-math-letters ]
4957   { encoding = OML,
4958     family   = cmm,
4959     series    = {m,b},
4960     shape     = it }
4961   {
4962     A = {100, 50}, % \mathnormal
4963     B = { 50,   },
4964     C = { 50,   },
4965     D = { 50, 50},
4966     E = { 50,   },
4967     F = {100, 50},
4968     G = { 50, 50},
4969     H = { 50, 50},
4970     I = { 50, 50},
4971     J = {150, 50},
4972     K = { 50,100},
4973     L = { 50, 50},
4974     M = { 50,   },
4975     N = { 50,   },
4976     O = { 50,   },
4977     P = { 50,   },
4978     Q = { 50, 50},
4979     R = { 50,   },
4980     S = { 50,   },
4981     T = { 50,100},
4982     U = { 50, 50},
4983     V = {100,100},
4984     W = { 50,100},
4985     X = { 50,100},
4986     Y = {100,100},
4987     f = {100,100},
4988     h = {   ,100},
4989     i = {   , 50},
4990     j = {   , 50},
4991     k = {   , 50},
4992     r = {   , 50},
4993     v = {   , 50},
4994     w = {   , 50},
4995     x = {   , 50},
4996     "0B = { 50,100}, % \alpha
4997     "0C = { 50, 50}, % \beta
4998     "0D = {200,150}, % \gamma
4999     "0E = { 50, 50}, % \delta
5000     "0F = { 50, 50}, % \epsilon
5001     "10 = { 50,150}, % \zeta
5002 %    "11 = {   ,   }, % \eta
5003     "12 = { 50,   }, % \theta
5004     "13 = {   ,100}, % \iota
5005     "14 = {   ,100}, % \kappa
5006     "15 = {100, 50}, % \lambda
```

```

5007 "16 = { , 50}, % \mu
5008 "17 = { , 50}, % \nu
5009 "18 = { , 50}, % \xi
5010 "19 = { 50,100}, % \pi
5011 "1A = { 50, 50}, % \rho
5012 "1B = { ,150}, % \sigma
5013 "1C = { 50,150}, % \tau
5014 "1D = { 50, 50}, % \upsilon
5015 % "1E = { , }, % \phi
5016 "1F = { 50,100}, % \chi
5017 "20 = { 50, 50}, % \psi
5018 "21 = { , 50}, % \omega
5019 "22 = { , 50}, % \varepsilon
5020 "23 = { , 50}, % \vartheta
5021 "24 = { , 50}, % \varpi
5022 "25 = {100, }, % \varrho
5023 "26 = {100,100}, % \varsigma
5024 "27 = { 50, 50}, % \varphi
5025 "28 = {100,100}, % \leftharpoonup
5026 "29 = {100,100}, % \leftharpoondown
5027 "2A = {100,100}, % \rightharpoonup
5028 "2B = {100,100}, % \rightharpoondown
5029 "2C = {300,200}, % \lhook
5030 "2D = {200,300}, % \rhook
5031 "2E = { ,100}, % \triangleright
5032 "2F = {100, }, % \triangleleft
5033 % 0 - 9
5034 "3A = { ,500}, % ., \ldotp
5035 "3B = { ,500}, % ,
5036 "3C = {200,100}, % <
5037 "3D = {300,400}, % /
5038 "3E = {100,200}, % >
5039 "3F = {200,200}, % \star
5040 % "40 = { , }, % \partial
5041 "5B = { ,100}, % \flat
5042 % "5C = { , }, % \natural
5043 % "5D = { , }, % \sharp
5044 "5E = {200,200}, % \smile
5045 "5F = {200,200}, % \frown
5046 % "60 = { , }, % \ell
5047 % "7B = { , }, % \imath
5048 "7C = {100, }, % \jmath
5049 "7D = { ,100}, % \wp
5050 }
5051

```

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}

```

```

5052 \SetProtrusion
5053 [ name = cmr-math-symbols ]
5054 { encoding = OMS,
5055   family = cmsy,
5056   series = {m,b},
5057   shape = n }
5058 {
5059   A = {150, 50}, % \mathcal
5060   C = { ,100},
5061   D = { , 50},

```



```

5062     F = { 50,150},
5063     I = {    ,100},
5064     J = {100,150},
5065     K = {    ,100},
5066     L = {100,   },
5067     M = { 50, 50},
5068     N = { 50,100},
5069     P = {    , 50},
5070     Q = { 50,   },
5071     R = {    , 50},
5072     T = { 50,150},
5073     V = { 50, 50},
5074     W = {    , 50},
5075     X = {100,100},
5076     Y = {100,   },
5077     Z = {100,150},
5078     "00 = {300,300}, % -
5079     "01 = {    ,700}, % \cdot, \cdotp
5080     "02 = {150,250}, % \times
5081     "03 = {150,250}, % *, \ast
5082     "04 = {200,300}, % \div
5083     "05 = {150,250}, % \diamond
5084     "06 = {200,200}, % \pm
5085     "07 = {200,200}, % \mp
5086     "08 = {100,100}, % \oplus
5087     "09 = {100,100}, % \ominus
5088     "0A = {100,100}, % \otimes
5089     "0B = {100,100}, % \oslash
5090     "0C = {100,100}, % \odot
5091     "0D = {100,100}, % \bigcirc
5092     "0E = {100,100}, % \circ
5093     "0F = {100,100}, % \bullet
5094     "10 = {100,100}, % \asymp
5095     "11 = {100,100}, % \equiv
5096     "12 = {200,100}, % \subseteq
5097     "13 = {100,200}, % \supseteq
5098     "14 = {200,100}, % \leq
5099     "15 = {100,200}, % \geq
5100     "16 = {200,100}, % \preceq
5101     "17 = {100,200}, % \succeq
5102     "18 = {200,200}, % \sim
5103     "19 = {150,150}, % \approx
5104     "1A = {200,100}, % \subset
5105     "1B = {100,200}, % \supset
5106     "1C = {200,100}, % \ll
5107     "1D = {100,200}, % \gg
5108     "1E = {300,100}, % \prec
5109     "1F = {100,300}, % \succ
5110     "20 = {100,200}, % \leftarrow
5111     "21 = {200,100}, % \rightarrow
5112     "22 = {100,100}, % \uparrow
5113     "23 = {100,100}, % \downarrow
5114     "24 = {100,100}, % \leftrightarrows
5115     "25 = {100,100}, % \nearrow
5116     "26 = {100,100}, % \searrow
5117     "27 = {100,100}, % \simeq
5118     "28 = {100,100}, % \Leftarrow
5119     "29 = {100,100}, % \Rightarrow
5120     "2A = {100,100}, % \Uparrow
5121     "2B = {100,100}, % \Downarrow

```

```

5122 "2C = {100,100}, % \Leftrightarrow
5123 "2D = {100,100}, % \nwarrow
5124 "2E = {100,100}, % \swarrow
5125 "2F = { ,100}, % \propto
5126 "30 = { ,400}, % \prime
5127 "31 = {100,100}, % \infty
5128 "32 = {150,100}, % \in
5129 "33 = {100,150}, % \ni
5130 "34 = {100,100}, % \triangle, \bigtriangleup
5131 "35 = {100,100}, % \bigtriangledown
5132 % "36 = { , }, % \not
5133 % "37 = { , }, % \mapstochar
5134 "38 = { ,100}, % \forall
5135 "39 = {100, }, % \exists
5136 "3A = {200, }, % \neg
5137 % "3B = { , }, % \emptyset
5138 % "3C = { , }, % \Re
5139 % "3D = { , }, % \Im
5140 "3E = {200,200}, % \top
5141 "3F = {200,200}, % \bot, \perp
5142 % "40 = { , }, % \aleph
5143 % "5B = { , }, % \cup
5144 % "5C = { , }, % \cap
5145 % "5D = { , }, % \uplus
5146 "5E = {100,200}, % \wedge
5147 "5F = {100,200}, % \vee
5148 "60 = { ,300}, % \vdash
5149 "61 = {300, }, % \dashv
5150 "62 = {100,100}, % \lfloor
5151 "63 = {100,100}, % \rfloor
5152 "64 = {100,100}, % \lceil
5153 "65 = {100,100}, % \rceil
5154 "66 = {150, }, % \lbrace
5155 "67 = { ,150}, % \rbrace
5156 "68 = {400, }, % \langle
5157 "69 = { ,400}, % \rangle
5158 % "6A = { , }, % \arrowvert, \mid, \vert, |
5159 % "6B = { , }, % \Arrowvert, \parallel, \Vert
5160 "6C = {100,100}, % \updownarrow
5161 "6D = {100,100}, % \Updownarrow
5162 "6E = {100,300}, % \, \backslash, \setminus
5163 % "6F = { , }, % \wr
5164 % "70 = { , }, % \sqrt{\phantom{x}}
5165 % "71 = { , }, % \amalg
5166 "72 = {100,100}, % \nabla
5167 % "73 = { , }, % \smallint
5168 % "74 = { , }, % \sqcup
5169 % "75 = { , }, % \sqcap
5170 % "76 = { , }, % \sqsubseteq
5171 % "77 = { , }, % \sqsupseteq
5172 % "78 = { , }, % \mathsection
5173 "79 = {200,200}, % \dagger
5174 "7A = {100,100}, % \ddagger
5175 "7B = {100, }, % \mathparagraph
5176 "7C = {100,100}, % \clubsuit
5177 "7D = {100,100}, % \diamondsuit
5178 "7E = {100,100}, % \heartsuit
5179 "7F = {100,100}, % \spadesuit
5180 }
5181

```

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}
```

```
5182%\SetProtrusion
5183% [ name      = cmr-math-largesymbols ]
5184% { encoding = OMX,
5185%   family   = {cmex,cmr} }
5186% {
5187%   "00 % (
5188%   "01 % )
5189%   "02 % [
5190%   "03 % ]
5191%   "04 % \lfloor
5192%   "05 % \rfloor
5193%   "06 % \lceil
5194%   "07 % \rceil
5195%   "08 % \lbrace
5196%   "09 % \rbrace
5197%   "0A % <, \angle
5198%   "0B % >, \angle
5199%   "0C % \vert, |
5200%   "0D % \Vert
5201%   "0E % /
5202%   "0F % \backslash
5203%   "3A % \lggroup
5204%   "3B % \rgroup
5205%   "3C % \arrowvert
5206%   "3D % \Arrowvert
5207%   "3E % \bracevert
5208%   "3F % \updownarrow
5209%   "40 % \lmoustache
5210%   "41 % \rmoustache
5211%   "46 % \bigsqcup
5212%   "48 % \oint
5213%   "4A % \bigodot
5214%   "4C % \bigoplus
5215%   "4E % \bigotimes
5216%   "50 % \sum
5217%   "51 % \prod
5218%   "52 % \intop
5219%   "53 % \bigcup
5220%   "54 % \bigcap
5221%   "55 % \biguplus
5222%   "56 % \bigwedge
5223%   "57 % \bigvee
5224%   "60 % \coprod
5225%   "70 % \sqrt{\text{sign}}
5226%   "77 % \Updownarrow
5227%   "78 % \uparrow
5228%   "79 % \downarrow
5229%   "7A % \braced, \lmoustache
5230%   "7B % \bracerd, \rmoustache
5231%   "7C % \bracelu
5232%   "7D % \braceru
5233%   "7E % \Uparrow
5234%   "7F % \Downarrow
5235% }
5236</cmr>
```

5237 *<!/cfg-u>*

### 14.6.7 AMS fonts

5238 *<\*cfg-u>*

Symbol font ‘a’, defined in amssymb.

```
\DeclareSymbolFont{AMSA}{U}{msa}{m}{n}
```

```
5239 <*msa>
5240 \SetProtrusion
5241 [ name = AMSa ]
5242 { encoding = U,
5243   family = msa }
5244 {
5245 % "00 = { , }, % \boxdot
5246 % "01 = { , }, % \boxplus
5247 % "02 = { , }, % \boxtimes
5248 % "03 = { , }, % \square
5249 % "04 = { , }, % \blacksquare
5250 % "05 = {150,250}, % \centerdot
5251 % "06 = {100,100}, % \lozenge
5252 % "07 = { 50, 50}, % \blacklozenge
5253 % "08 = { 50, 50}, % \circlearrowright
5254 % "09 = { 50, 50}, % \circlearrowleft
5255 % "0A = { 50, 50}, % \rightleftharpoons
5256 % "0B = { 50, 50}, % \leftrightharpoons
5257 % "0C = { , }, % \boxminus
5258 % "0D = {-50,150}, % \Vdash
5259 % "0E = {-50,150}, % \Vvdash
5260 % "0F = {-70,150}, % \vDash
5261 % "10 = {100,100}, % \twoheadrightarrow
5262 % "11 = { 50,150}, % \twoheadleftarrow
5263 % "12 = { 50,100}, % \leftleftarrows
5264 % "13 = { 50, 80}, % \rightrightarrows
5265 % "14 = {120,120}, % \upuparrows
5266 % "15 = {120,120}, % \downdownarrows
5267 % "16 = {200,200}, % \upharpoonright
5268 % "17 = {200,200}, % \downharpoonright
5269 % "18 = {200,200}, % \upharpoonleft
5270 % "19 = {200,200}, % \downharpoonleft
5271 % "1A = { 50, 80}, % \rightarrowtail
5272 % "1B = { 50, 80}, % \leftarrowtail
5273 % "1C = { 50, 50}, % \leftrightarrows
5274 % "1D = { 50, 50}, % \rightleftarrows
5275 % "1E = {150, }, % \Lsh
5276 % "1F = { ,150}, % \Rsh
5277 % "20 = { 50,100}, % \rightsquigarrow
5278 % "21 = { 50, 50}, % \leftrightsquigarrow
5279 % "22 = { 50, 50}, % \looparrowleft
5280 % "23 = { 50, 50}, % \looparrowright
5281 % "24 = { , 50}, % \circeq
5282 % "25 = { ,100}, % \succsim
5283 % "26 = { ,100}, % \gtrsim
5284 % "27 = { ,100}, % \gtrapprox
5285 % "28 = { 50, 50}, % \multimap
5286 % "29 = { , }, % \therefore
5287 % "2A = { , }, % \because
5288 % "2B = { 50, 50}, % \doteqdot
```

```

5289 "2C = { 50, 50}, % \triangleq
5290 "2D = {100, 50}, % \preccurlyeq
5291 "2E = {100, 50}, % \lessapprox
5292 "2F = { 50, 50}, % \lessapprox
5293 "30 = {100, 50}, % \eqslantless
5294 "31 = { 50, 50}, % \eqslantgtr
5295 "32 = {100, 50}, % \curlyeqprec
5296 "33 = { 50,100}, % \curlyeqsucc
5297 "34 = {100, 50}, % \preccurlyeq
5298 % "35 = { , }, % \leqq
5299 "36 = { 50, }, % \leqslant
5300 % "37 = { , }, % \lessgtr
5301 "38 = { , 50}, % \backprime
5302 "39 = {200,200}, % \dabar@ : the dash bar in \dash(left,right)arrow
5303 % "3A = { , }, % \risingdotseq
5304 % "3B = { , }, % \fallingdotseq
5305 "3C = { 50,100}, % \succcurlyeq
5306 % "3D = { , }, % \geqq
5307 "3E = { , 50}, % \geqslant
5308 % "3F = { , }, % \gtrless
5309 "40 = { , 50}, % \sqsubset
5310 "41 = { 50, }, % \sqsupset
5311 "42 = { ,150}, % \vartriangleright, \rhd
5312 "43 = {150, }, % \vartriangleleft, \lhd
5313 "44 = { ,100}, % \trianglerighteq, \unrhd
5314 "45 = {100, }, % \trianglelefteq, \unlhd
5315 "46 = {100,100}, % \bigstar
5316 % "47 = { , }, % \between
5317 "48 = { 50, 50}, % \blacktriangledown
5318 "49 = { ,100}, % \blacktriangleright
5319 "4A = {100, }, % \blacktriangleleft
5320 "4B = { ,150}, % \dashrightarrow (the arrow)
5321 "4C = {150, }, % \dashleftarrow
5322 "4D = { 50, 50}, % \vartriangle
5323 "4E = { 50, 50}, % \blacktriangle
5324 "4F = { 50, 50}, % \triangledown
5325 "50 = { 50, 50}, % \eqcirc
5326 % "51 = { , }, % \lesseqgtr
5327 % "52 = { , }, % \gtreqless
5328 % "53 = { , }, % \lesseqgtr
5329 % "54 = { , }, % \gtreqless
5330 % "55 = { , }, % \yen
5331 "56 = { ,150}, % \Rrightarrow
5332 "57 = {150, }, % \Lleftarrow
5333 "58 = {100,200}, % \checkmark
5334 % "59 = { , }, % \veebar
5335 % "5A = { , }, % \barwedge
5336 % "5B = { , }, % \doublebarwedge
5337 "5C = { 50, 50}, % \angle
5338 "5D = { 50, 50}, % \measuredangle
5339 "5E = { 50, 50}, % \sphericalangle
5340 "5F = { , 50}, % \varpropto
5341 "60 = {100,100}, % \smallsmile
5342 "61 = {100,100}, % \smallfrown
5343 "62 = { 50, }, % \Subset
5344 "63 = { , 50}, % \Supset
5345 % "64 = { , }, % \Cup
5346 % "65 = { , }, % \Cap
5347 "66 = {100,100}, % \curlywedge
5348 "67 = {100,100}, % \curlyvee

```

```

5349 "68 = { 50,100}, % \leftthreetimes
5350 "69 = {100, 50}, % \rightthreetimes
5351 % "6A = { , }, % \subseteqq
5352 % "6B = { , }, % \supseteqq
5353 "6C = { 50, 50}, % \bumpeq
5354 "6D = { 50, 50}, % \Bumpeq
5355 "6E = {100, }, % \lll
5356 "6F = { ,100}, % \ggg
5357 "70 = { 50,100}, % \ulcorner
5358 "71 = {100, 50}, % \urcorner
5359 % "72 = { , }, % \circledR
5360 % "73 = { , }, % \circledS
5361 % "74 = { , }, % \pitchfork
5362 "75 = {100,150}, % \dotplus
5363 "76 = { 50,100}, % \backsimeq
5364 % "77 = { , }, % \backsimeq
5365 "78 = { 50,100}, % \llcorner
5366 "79 = {100, 50}, % \lrcorner
5367 % "7A = { , }, % \maltese
5368 % "7B = { , }, % \complement
5369 "7C = {100,100}, % \intercal
5370 "7D = { 50, 50}, % \circledcirc
5371 "7E = { 50, 50}, % \circledast
5372 "7F = { 50, 50}, % \circleddash
5373 }
5374
5375 </msa>

```

Symbol font ‘b’.

```

\DeclareSymbolFont{AMSb}{U}{msb}{m}{n}
\DeclareSymbolFontAlphabet{\mathbb}{AMSb}

```

```

5376 <msb>
5377 \SetProtrusion
5378 [ name = AMSb ]
5379 { encoding = U,
5380   family = msb }
5381 {
5382   A = { 50, 50}, % \mathbb
5383   C = { 50, 50},
5384   G = { , 50},
5385   L = { , 50},
5386   P = { , 50},
5387   R = { , 50},
5388   T = { , 50},
5389   V = { 50, 50},
5390   X = { 50, 50},
5391   Y = { 50, 50},
5392   "00 = { 50, 50}, % \lvertneqq
5393   "01 = { 50, 50}, % \gvertneqq
5394   "02 = { 50, 50}, % \nleq
5395   "03 = { 50, 50}, % \ngeq
5396   "04 = {100, 50}, % \nless
5397   "05 = { 50,150}, % \ngtr
5398   "06 = {100, 50}, % \nprec
5399   "07 = { 50,150}, % \nsucc
5400   "08 = { 50, 50}, % \lneqq
5401   "09 = { 50, 50}, % \gneqq
5402   "0A = {100,100}, % \nleqslant
5403   "0B = {100,100}, % \ngeqslant

```

```

5404 "0C = {100, 50}, % \lneq
5405 "0D = { 50,100}, % \gneq
5406 "0E = {100, 50}, % \npreceq
5407 "0F = { 50,100}, % \nsucceq
5408 "10 = { 50,  }, % \precnsim
5409 "11 = { 50, 50}, % \succnsim
5410 "12 = { 50, 50}, % \lnsim
5411 "13 = { 50, 50}, % \gnsim
5412 "14 = { 50, 50}, % \nleqq
5413 "15 = { 50, 50}, % \ngeqq
5414 "16 = { 50, 50}, % \precneqq
5415 "17 = { 50, 50}, % \succneqq
5416 "18 = { 50, 50}, % \precnapprox
5417 "19 = { 50, 50}, % \succnapprox
5418 "1A = { 50, 50}, % \lnapprox
5419 "1B = { 50, 50}, % \gnapprox
5420 "1C = {150,200}, % \nsim
5421 "1D = { 50, 50}, % \ncong
5422 "1E = {100,150}, % \diagup
5423 "1F = {100,150}, % \diagdown
5424 "20 = {100, 50}, % \varsubsetneq
5425 "21 = { 50,100}, % \varsupsetneq
5426 "22 = {100, 50}, % \subsetneqq
5427 "23 = { 50,100}, % \supsetneqq
5428 "24 = {100, 50}, % \subsetneqq
5429 "25 = { 50,100}, % \supsetneqq
5430 "26 = {100, 50}, % \varsubsetneqq
5431 "27 = { 50,100}, % \varsupsetneqq
5432 "28 = {100, 50}, % \subsetneq
5433 "29 = { 50,100}, % \supsetneq
5434 "2A = {100, 50}, % \subseteq
5435 "2B = { 50,100}, % \supseteq
5436 "2C = { 50,100}, % \nparallel
5437 "2D = {100,150}, % \nmid
5438 "2E = {150,150}, % \nshortmid
5439 "2F = {100,100}, % \nshortparallel
5440 "30 = {  ,150}, % \nvDash
5441 "31 = {  ,150}, % \nVDash
5442 "32 = {  ,100}, % \nvDash
5443 "33 = {  ,100}, % \nVDash
5444 "34 = {  ,100}, % \ntrianglerighteq
5445 "35 = {100,  }, % \ntrianglelefteq
5446 "36 = {100,  }, % \ntriangleleft
5447 "37 = {  ,100}, % \ntriangleright
5448 "38 = {100,200}, % \leftarrow
5449 "39 = {100,200}, % \rightarrow
5450 "3A = {100,100}, % \Leftarrow
5451 "3B = { 50,100}, % \Rightarrow
5452 "3C = {100,100}, % \Leftrightarrow
5453 "3D = {100,200}, % \leftrightharpoonup
5454 "3E = { 50, 50}, % \divideontimes
5455 "3F = { 50, 50}, % \varnothing
5456 % "40 = {  ,  }, % \nexists
5457 "60 = {200,  }, % \Finv
5458 "61 = {  , 50}, % \Game
5459 % "66 = {  ,  }, % \mho
5460 % "67 = {  ,  }, % \eth
5461 "68 = {100,100}, % \eqsim
5462 "69 = { 50,  }, % \beth
5463 "6A = { 50,  }, % \gimel

```

```

5464 "6B = {150, }, % \daleth
5465 "6C = {200, }, % \lessdot
5466 "6D = { ,200}, % \gtrdot
5467 "6E = {100,200}, % \ltimes
5468 "6F = {150,100}, % \rtimes
5469 "70 = { 50,100}, % \shortmid
5470 "71 = { 50, 50}, % \shortparallel
5471 "72 = {200,300}, % \smallsetminus
5472 "73 = {100,200}, % \thicksim
5473 "74 = { 50,100}, % \thickapprox
5474 "75 = { 50, 50}, % \approx
5475 "76 = { 50,100}, % \succapprox
5476 "77 = { 50, 50}, % \precapprox
5477 "78 = {100,100}, % \curvearrowleft
5478 "79 = { 50,150}, % \curvearrowright
5479 "7A = { 50,200}, % \digamma
5480 "7B = {100, 50}, % \varkappa
5481 % "7C = { , }, % \Bbbk
5482 % "7D = { , }, % \hslash
5483 % "7E = { , }, % \hbar
5484 "7F = {200, }, % \backslashepsilon
5485 }
5486
5487 </msb>

```

Euler Fraktur font (eufrak).

```

\DeclareMathAlphabet{\mathfrak}{U}{euf}{m}{n}
\SetMathAlphabet{\mathfrak}{bold}{U}{euf}{b}{n}

```

```

5488 <*euf>
5489 \SetProtrusion
5490 [ name = mathfrak ]
5491 { encoding = U,
5492   family = euf }
5493 {
5494   A = { , 50},
5495   B = { , 50},
5496   C = { 50, 50},
5497   D = { , 80},
5498   E = { 50, },
5499   G = { , 50},
5500   L = { , 80},
5501   O = { , 50},
5502   T = { , 80},
5503   X = { 80, 50},
5504   Z = { 80, 50},
5505   b = { , 50},
5506   c = { , 50},
5507   k = { , 50},
5508   p = { , 50},
5509   q = { 50, },
5510   v = { , 50},
5511   w = { , 50},
5512   x = { , 50},
5513   1 = {100,100},
5514   2 = { 80, 80},
5515   3 = { 80, 50},
5516   4 = { 80, 50},
5517   7 = { 50, 50},
5518 }

```



5519

5520  $\langle /euf \rangle$ 

Euler script font (euca1).

```
\DeclareMathAlphabet\EuScript{U}{eus}{m}{n}
\SetMathAlphabet\EuScript{bold}{U}{eus}{b}{n}
```

5521  $\langle *eus \rangle$ 

5522 \SetProtrusion

5523 [ name = euscript ]

5524 { encoding = U,

5525 family = eus }

5526 {

5527 A = {100,100},

5528 B = { 50,100},

5529 C = { 50, 50},

5530 D = { 50,100},

5531 E = { 50,100},

5532 F = { 50, },

5533 G = { 50, },

5534 H = { ,100},

5535 K = { , 50},

5536 L = { ,150},

5537 M = { , 50},

5538 N = { , 50},

5539 O = { 50, 50},

5540 P = { 50, 50},

5541 T = { ,100},

5542 U = { , 50},

5543 V = { 50, 50},

5544 W = { 50, 50},

5545 X = { 50, 50},

5546 Z = { 50,100},

5547 }

5548

5549  $\langle /eus \rangle$ 5550  $\langle /cfg-u \rangle$ 5551  $\langle *beta \rangle$ 

## 14.7 Interword Spacing

Default unit is space.

5552 %%% -----

5553 %%% INTERWORD SPACING SETTINGS

5554

5555 \SetExtraSpacing

5556 [ name = default ]

5557 { encoding = {OT1,T1,LY1,OT4,T5} }

5558 {

These settings are only a first approximation. The following reasoning is from a mail from Ulrich Dirr. 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

5559                   {,} = { , -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)

5560                   r = { , -300, 300},

- before or after lowercase characters with ascenders

5561                   b = { , -200, 200},

5562                   d = { , -200, 200},

5563                   f = { , -200, 200},

5564                   h = { , -200, 200},

5565                   k = { , -200, 200},

5566                   l = { , -200, 200},

5567                   t = { , -200, 200},

- before of after lowercase characters with x-height plus descender with additional optical space, e. g., 'v', or 'w'

5568                   c = { , -100, 100},

5569                   p = { , -100, 100},

5570                   v = { , -100, 100},

5571                   w = { , -100, 100},

5572                   z = { , -100, 100},

5573                   x = { , -100, 100},

5574                   y = { , -100, 100}, % ?

- before of after lowercase characters with x-height plus descender without additional optical space

5575                   i = { , 50, -50},

5576                   m = { , 50, -50},

5577                   n = { , 50, -50},

5578                   u = { , 50, -50},

- after colon and semicolon

5579                   : = { , 200, -200},

5580                   ; = { , 200, -200},

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

5581                   . = { , 250, -250},

5582                   ! = { , 250, -250},

5583                   ? = { , 250, -250},

The order has to be reversed when enlarging is needed.'

5584                   }

5585

Questions are:

- Is the result really better?

- Is it overdone? (Try with a factor  $< 1000$ .)
- Should the first parameter also be used? (Probably.)

The following settings simulate `\nonfrenchspacing` (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the  $\TeX$ book:

‘If the space factor  $f$  is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if  $f \geq 2000$ . [...] Then the stretch component is multiplied by  $f / 1000$ , while the shrink component is multiplied by  $1000 / f$ .’

The ‘extra space’ (`\fontdimen7`) for Computer Modern Roman is a third of `\fontdimen2`, i. e., 333.

```
5586 \SetExtraSpacing
5587   [ name      = nonfrench-cmr,
5588     load      = default,
5589     context   = nonfrench ]
5590   { encoding = {OT1,T1,LY1,OT4,T5},
5591     family   = cmr }
5592   {
```

`latex.ltx` has:

```
\def\nonfrenchspacing{
\sffcode'\. 3000
```

```
5593   . = {333,2000,-667},
```

```
\sffcode'\? 3000
```

```
5594   ? = {333,2000,-667},
```

```
\sffcode'\! 3000
```

```
5595   ! = {333,2000,-667},
```

```
\sffcode'\: 2000
```

```
5596   : = {333,1000,-500},
```

```
\sffcode'\; 1500
```

```
5597   ; = { , 500,-333},
```

```
\sffcode'\, 1250
```

```
5598   {,}= { , 250,-200},
```

```
}
```

```
5599 }
```

```
5600
```

fontinst, however, which is also used to create the PSNFSS font metrics, sets it to 240 by default. Therefore, the fallback settings use this value for the first component.

```

5601 \SetExtraSpacing
5602   [ name      = nonfrench-default,
5603     load       = default,
5604     context    = nonfrench ]
5605   { encoding = {OT1,T1,LY1,OT4,T5} }
5606   {
5607     . = {240,2000,-667},
5608     ? = {240,2000,-667},
5609     ! = {240,2000,-667},
5610     : = {240,1000,-500},
5611     ; = {    , 500,-333},
5612     {,}= {    , 250,-200},
5613   }
5614
```

## 14.8 Additional Kerning

Default unit is 1em.

```

5615 %%% -----
5616 %%% ADDITIONAL KERNING
5617
```

A dummy list to be loaded when no context is active.

```

5618 \SetExtraKerning
5619   [ name = empty ]
5620   { encoding = {OT1,T1,LY1,OT4,T5,TS1} }
5621   { }
5622
5623 \SetExtraKerning
5624   [ name      = french-default,
5625     context    = french,
5626     unit       = space ]
5627   { encoding = {OT1,T1,LY1} }
5628   {
5629     : = {1000,}, % = \fontdimen2
5630     ; = {500,}, % ~ \thinspace
5631     ! = {500,},
5632     ? = {500,},
5633   }
5634
```

This has the disadvantage that the word following a left guillemot will not be hyphenated. This might be fixed in pdfTeX.

```

5635 \SetExtraKerning
5636   [ name      = french-guillemets,
5637     context    = french-guillemets,
5638     load       = french-default,
5639     unit       = space ]
5640   { encoding = {OT1,T1,LY1} }
5641   {
5642     \guillemotleft = { ,800}, % = 0.8\fontdimen2
5643     \guillemotright = {800, },
5644   }
5645
```

```

5646 \SetExtraKerning
5647   [ name      = turkish,
5648     context   = turkish,
5649     unit      = space ]
5650   { encoding = {OT1,T1,LY1} }
5651   {
5652     : = {500, }, % ~ \thinspace
5653     ! = {500, },
5654     {=} = {500, },
5655   }
5656

```

The settings with the ‘letterspacing’ context will be loaded whenever the command `\textls` resp. `\lsstyle` are used.

```

5657 %% The following settings with 'context=letterspacing'
5658 %% will apply to \lsstyle and \textls:
5659 \SetExtraKerning
5660   [ name      = letterspacing-default,
5661     context   = letterspacing,
5662     unit      = 1em,
5663     preset    = {1000,1000} ]
5664   { encoding = {OT1,OT4} }
5665   {

```

The full stop and quotation marks should be spaced out less. Numbers are not spaced out, according to soul.

```

5666     . = {0, },
5667     0 = {0,0},
5668     1 = {0,0},
5669     2 = {0,0},
5670     3 = {0,0},
5671     4 = {0,0},
5672     5 = {0,0},
5673     6 = {0,0},
5674     7 = {0,0},
5675     8 = {0,0},
5676     9 = {0,0},
5677     \textquoteleft = {0,0}, \textquoteright = {0,0},
5678     \textquotedblleft = {0,0}, \textquotedblright = {0,0},
5679   }
5680
5681 \SetExtraKerning
5682   [ name      = letterspacing-T1,
5683     load      = letterspacing-default,
5684     context   = letterspacing,
5685     unit      = 1em,
5686     preset    = {1000,1000} ]
5687   { encoding = {T1,LY1,T5} }
5688   {
5689     \quotesinglbase = {0,0}, \quotedblbase = {0,0},
5690     \guilsinglleft = {0,0}, \guilsinglright = {0,0},
5691     \guillemotleft = {0,0}, \guillemotright = {0,0},
5692   }
5693
5694 /beta
5695 /config

```

## 15 Auxiliary File for Micro Fine Tuning

This file can be used to test protrusion and expansion settings.

```

5696 (*test)
5697 \documentclass{article}
5698
5699 %% Here you can specify the font you want to test, using
5700 %% the commands \fontfamily, \fontseries, and \fontshape.
5701 %% Make sure to end all lines with a comment character!
5702 \newcommand*\TestFont{%
5703   \fontfamily{ppl}%
5704   \fontseries{b}%
5705   \fontshape{it}% sc, sl
5706 }
5707
5708 \usepackage{ifthen}
5709 \usepackage[T1]{fontenc}
5710 \usepackage[latin1]{inputenc}
5711 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
5712
5713 \pagestyle{empty}
5714 \setlength{\parindent}{0pt}
5715 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
5716 \newcommand*\testprotrusion[2][{}]{%
5717   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
5718   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
5719   \ifthenelse{\equal{#1}{l}}{\crulefill}{\leftarrowfill} #2
5720   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
5721   you know the rest%
5722   \ifthenelse{\equal{#1}{l}}{\crulefill}{\leftarrowfill} #2
5723   \linebreak
5724   {\fontencoding{\encodingdefault}%
5725    \fontseries{\seriesdefault}%
5726    \fontshape{\shapedefault}%
5727    \selectfont
5728    Here is the beginning of a line, \dotfill and here is its end}\linebreak
5729 }
5730 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
5731 \def\stripprefix#1>{}
5732 \newcount\charcount
5733 \begin{document}
5734
5735 \microtypesetup{expansion=false}
5736
5737 {\centering The font in this document is called by:\\
5738 \texttt{\showTestFont}\par}\bigskip
5739
5740 \TestFont\selectfont
5741 This line intentionally left empty\linebreak
5742 %% A -- Z
5743 \charcount=65
5744 \loop
5745   \testprotrusion{\char\charcount}
5746   \advance\charcount 1
5747   \ifnum\charcount < 91 \repeat
5748 %% a -- z
5749 \charcount=97
5750 \loop
5751   \testprotrusion{\char\charcount}

```

```

5752 \advance\charcount 1
5753 \ifnum\charcount < 123 \repeat
5754 %% 0 -- 9
5755 \charcount=48
5756 \loop
5757 \testprotrusion{\char\charcount}
5758 \advance\charcount 1
5759 \ifnum\charcount < 58 \repeat
5760 %%
5761 \testprotrusion[r]{,}
5762 \testprotrusion[r]{.}
5763 \testprotrusion[r]{;}
5764 \testprotrusion[r]{:}
5765 \testprotrusion[r]{?}
5766 \testprotrusion[r]{!}
5767 \testprotrusion[l]{\textexclamdown}
5768 \testprotrusion[l]{\textquestiondown}
5769 \testprotrusion[r]{})}
5770 \testprotrusion[l]{(}
5771 \testprotrusion{/}
5772 \testprotrusion{\char'\}
5773 \testprotrusion{-}
5774 \testprotrusion{\textendash}
5775 \testprotrusion{\textemdash}
5776 \testprotrusion{\textquoteleft}
5777 \testprotrusion{\textquoteright}
5778 \testprotrusion{\textquotedblleft}
5779 \testprotrusion{\textquotedblright}
5780 \testprotrusion{\quotesinglbase}
5781 \testprotrusion{\quotedblbase}
5782 \testprotrusion{\guilsinglleft}
5783 \testprotrusion{\guilsinglright}
5784 \testprotrusion{\guillemotleft}
5785 \testprotrusion{\guillemotright}
5786
5787 \newpage
5788 The following displays the current font stretched by 5%,
5789 normal, and shrunk by 5\%:
5790
5791 \bigskip
5792 \newlength{\MTln}
5793 \newcommand*{\teststring}
5794 {ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
5795 \settowidth{\MTln}{\teststring}
5796 \microtypesetup{expansion=true}
5797
5798 \parbox{1.05\MTln}{\teststring\linebreak\}
5799 \teststring\par\bigskip
5800 \parbox{0.95\MTln}{\teststring}
5801
5802 \end{document}
5803 /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 Harald Harders) . . . . .	57	family has already been loaded . . . . .	57
issue an error instead of a warning, when pdfTeX version is too old for autoexpand . . . . .	102	\MT@get@basefamily: only remove suffix, if it is 'x' or 'j' . . . . .	58
remove 8-bit characters from the configuration files (suggested by Harald Harders) . . . . .	108	\MT@get@listname@: don't check for empty characteristics list . . . . .	58
Protrusion: add factors for some more characters . . . . .	112	\MT@ifempty: bug fix: use category code 12 for the percent character (reported by Tom Kink) . . . . .	34
settings for Adobe Minion (contributed by Harald Harders) . . . . .	113	\MT@is@number: numbers may also be specified in hexadecimal or octal (suggested by Harald Harders) . . . . .	62
\DeclareCharacterInheritance: new macro: possibility to specify character inheritance . . . . .	85	\MT@pdftex@no: bug fix concerning version check (reported by Harald Harders) . . . . .	26
\MT@declare@sets: remove spaces around set name . . . . .	72	\MT@permute: don't use sets for empty encoding . . . . .	88
\MT@DeclareSet@: remove spaces around first argument . . . . .	71	\MT@pr@split: bug fix: allow zero and negative values . . . . .	43
\MT@find@file: bug fix: also check whether the file for the base font		\MT@use@set: remove spaces around set name . . . . .	76
		\UseMicrotypeSet: remove spaces around first argument . . . . .	76

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

General: check for packages that might load fonts . . . . .	66	as an alternative, not as a replacement . . . . .	40
Font Sets: declare cmor as an alias of cmr . . . . .	107	\MT@get@basefamily: also remove 'w' (swash capitals) . . . . .	58
new: allmath and basicmath . . . . .	106	\MT@get@highlevel: check whether defaults have changed . . . . .	73
Protrusion: add settings for Adobe Garamond and Computer Modern Roman in TS1 encoding . . . . .	130	\MT@get@listname@: alternatively check for alias font name . . . . .	58
add settings for Computer Modern Roman math symbols . . . . .	134	\MT@get@size: additional magic to catch some errors . . . . .	74
\MT@context: bug fix: set inheritance list globally to \empty . . . . .	60	hijack \set@fontsize instead of \setfontsize . . . . .	74
\MT@define@inh@key@encoding: check whether only one encoding specified . . . . .	86	\MT@get@slot: bug fix: group must also include \MT@get@composite . . . . .	61
\MT@familyalias: define alias font name		\MT@loop: bug fix: new macro, used instead of \loop . . . . .	37



\MT@maybe@do: also check for alias font name . . . . .	40	\MT@setupfont: also search for alias font file . . . . .	39
\MT@permute@@@@@: more sanity checks for \SetProtrusion and \SetExpansion . . . . .	89	bug fix: call \@@enc@update if necessary . . . . .	39

### Version 1.3 (2004/10/27)

Font Sets: declare aer, zer and hfor as an alias of cmr . . . . .	107	\MT@get@codes@name: bug fix: specifying load option does no longer require to give a name, too . . . . .	81
\MT@catcodes: check some category codes (compatibility with german) . . . . .	27	\MT@load@list: check whether list exists . . . . .	56

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

General: don't use scratch registers in global definitions . . . . .	61	\microtypesetup: bug fix: set the correct levels, and remember them; warning when enabling an option disabled in package options . . . . .	91
no need to check for packages that might load fonts anymore . . . . .	66	\MT@pdfcprot@error: check for pdfcprot . . . . .	29
use \pickup@font instead of \define@newfont as the hook for \MT@setupfont . . . . .	66	\MT@set@ex@opt: bug fix: specifying extra options does no longer require to give a name, too . . . . .	81
use one instead of five counters . . . . .	38		
Protrusion: tweak quote characters for cmr variants (OT1, T1, lmr) . . . . .	116		

### Version 1.4a (2004/11/17)

General: new option: final . . . . .	96	ing files (reported by Michael Hoppe) . . . . .	58
\MT@begin@catcodes: bug fix: reset some more catcodes when read-			

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

General: bug fix: set catcodes before reading global configuration file (reported by Christoph Bier) . . . . .	98	try alias family name if encoding failed . . . . .	41
new message if \pdfoutput is changed . . . . .	100	\MT@get@basefamily: bug fix: failed for font names of the form abczz (reported by Georg Verweyen) . . . . .	58
optimization: use less \csnames and \expandafters . . . . .	32	\MT@get@slot: don't define \MT@char globally (save stack problem) . . . . .	61
Protrusion: harmonize dashes in upshape and italic (cmr, pad, ppl) . . . . .	112	\MT@ifdimen: don't set \MT@count globally (save stack problem) . . . . .	34
slanted like italics . . . . .	119	\MT@use@set: don't use undeclared font sets . . . . .	76
\MT@checklist@family: bug fix: don't			

**Version 1.5 (2004/12/15)**

General: defaults: step: 4 (suggested by Hàn Thế Thành) . . . . .	97	<code>\DeclareMicrotypeAlias</code> : remove spaces around arguments . . . . .	77
defaults: calculate step as <code>min(stretch,shrink)/5</code> . . . . .	101	<code>\MT@begin@catcodes</code> : reset catcode of ‘=’ (compatibility with Turkish babel) . . . . .	58
defaults: turn off expansion for DVI output . . . . .	100	<code>\MT@catcodes</code> : reset catcode of ‘^’ (compatibility with chemsym) . . . . .	27
disable automatic expansion for DVI output . . . . .	102	<code>\MT@get@highlevel</code> : don’t test defaults if called after begin document . . . . .	73
new option: selected, by default false (suggested by Hàn Thế Thành) . . . . .	95	<code>\MT@scale@factor</code> : warning for factors outside limits . . . . .	45
Documentation: add note about <code>DVIoutput</code> option . . . . .	7	<code>\MT@scale@to@em</code> : don’t use <code>\lcode</code> and <code>\rcode</code> for the calculation . . . . .	44
add short history (section 12) . . . . .	21	<code>\MT@set@ex@codes</code> : allow non-selected font expansion . . . . .	49
Inheritance: remove <code>\ss</code> from T1 list, add <code>\DJ</code> . . . . .	108	<code>\MT@set@pr@codes</code> : adjust protrusion factors before setting the inheriting characters . . . . .	42
Protrusion: settings for Bitstream Character . . . . .	113		

**Version 1.6 (2005/01/24)**

General: defaults: turn off expansion for old pdf $\TeX$ versions . . . . .	98	tune CMR math letters (OML encoding) . . . . .	135
disable automatic expansion for old pdf $\TeX$ versions . . . . .	102	<code>\MT@def@num@opt</code> : test whether numeric options receive a number . . . . .	97
load a font, if none is active . . . . .	32	<code>\MT@get@charwd</code> : use e- $\TeX$ ’s <code>\fontcharwd</code> , if available . . . . .	44
new option: factor, by default 1000 . . . . .	97	<code>\MT@get@inh@list</code> : correct message if selected is false . . . . .	60
restructure dtx file . . . . .	106	<code>\MT@set@ex@codes</code> : introduce factor option . . . . .	49
test whether <code>\pickup@font</code> has changed . . . . .	68	<code>\MT@set@pr@codes</code> : introduce factor option . . . . .	42
use e- $\TeX$ ’s <code>\ifcsname</code> and <code>\ifdefined</code> if defined . . . . .	32	<code>\MT@use@set</code> : retain current set if new set is undeclared . . . . .	76
Protrusion: add italic uppercase Greek letters . . . . .	119	<code>\MT@vinfo</code> : new macro: used instead of <code>\ifMT@verbose</code> . . . . .	25
improve settings for numbers (pointed out by Peter Muthesius) . . . . .	114		

**Version 1.6a (2005/02/02)**

Documentation: add table of fonts with tailored protrusion settings . . . . .	15	ported by Bernard Gaulle) . . . . .	61
<code>\MT@get@slot</code> : completely redone, hopefully more robust (compatible with frenchpro; problem re-		<code>\MT@pdftex@no</code> : new macro . . . . .	26
		<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older pdf $\TeX$ versions . . . . .	49

**Version 1.7 (2005/03/23)**

General: \SetExpansion: bug fix: remove space after autoexpand ..	85	\MT@get@size: comparison with 1 to allow size smaller than 1 (suggested by Andreas Böhmann) .....	74
\SetExpansion: don't allow automatic expansion for old pdfTeX versions .....	85	\MT@get@slot: remove backslash hack	61
allow specification of size ranges (suggested by Andreas Böhmann)	73	test for \chardefed commands ...	61
modify \showhyphens .....	103	test whether \<encoding>\<..> is defined .....	61
new value for verbose option: errors .....	96	\MT@if@list@exists: don't define \MT@#1@c@name \globally, here and elsewhere .....	60
shorter command names .....	38	\MT@increment: use e-TeX's \numexpr if available .....	38
warning when running in draft mode .....	100	\MT@is@composite: new macro: construct command for composite character; no uncontrolled expansion .....	65
Documentation: add hint about compatibility .....	18	\MT@scale: new macro: use e-TeX's \numexpr if available .....	38
remove table of match order .....	12	\MT@set@ex@codes: two versions of this macro .....	49
Protrusion: fix: remove \ from OT1, add \textbackslash to T1 encoding	115	\MT@split@name: don't define \MT@encoding &c. \globally ...	40
\DeclareMicrotypeAlias: may also be used inside configuration files ..	77	\MT@test@ast: make it simpler .....	73
\LoadMicrotypeFile: new macro (suggested by Andreas Böhmann) ..	78	\MT@try@order: always check for size, too (suggested by Andreas Böhmann) .....	59
\Microtype@Hook: new macro for font package authors .....	98	bug fix: also check for //series//shape// (reported by Andreas Böhmann) .....	59
\microtypesetup: bug fix: warning also when setting to (no)compatibility .....	91	\MT@warn@code@too@large: new macro: type out maximum protrusion factor .....	46
\MT@begin@catcodes: also use inside configuration commands .....	58	\MT@warn@err: new macro: for verbose=errors .....	25
reset catcode of ':' (compatibility with french* packages) .....	58		
\MT@get@listname@: use \tfor (Andreas Böhmann's idea) .....	58		

**Version 1.8 (2005/06/23)**

General: \SetProtrusion: new option: unit .....	84	widow .....	13
if font substitution has occurred, set up the substitute font, not the selected one .....	67	add hint about error messages ...	19
new option: config to load a different main configuration file .....	98	Font Sets: add U encoding to allmath	106
new option: unit, by default character .....	97	declare pxx and txx as alias fonts of ppl and ptm .....	107
Documentation: add example for factor option .....	12	Inheritance: remove \DJ from T1 list (it's the same as \DH) .....	108
add example of how to get rid of a		Protrusion: add LY1 characters for Times .....	118
		settings for AMS math fonts ....	113
		verified settings for slanted Computer Modern Roman .....	124

<code>\DeclareMicrotypeAlias</code> : warning when overriding an alias font . . .	77	<code>\MT@led@kern</code> : character protrusion with <code>ledmac</code> . . . . .	29
<code>\DeclareMicrotypeSetDefault</code> : new macro: set fall back font set . . .	77	<code>\MT@make@string</code> : use <code>\@onelevel@sanitize</code> . . . . .	38
<code>\MT@begin@catcodes</code> : reset catcodes of the remaining ASCII characters . . .	58	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\@for</code> . . . . .	35
<code>\MT@check@rlist</code> : made recursive . . .	90	<code>\MT@map@tlist@n</code> : new macro: used instead of <code>\@tfor</code> . . . . .	36
<code>\MT@curr@list@name</code> : new macro: current list type and name . . . . .	65	<code>\MT@old@cmd</code> : renamed commands from <code>\..MicroType..</code> to <code>\..Microtype..</code> . . . . .	25
<code>\MT@declare@sets</code> : warning when redefining a set . . . . .	72	<code>\MT@orig@add@accent</code> : bug fix: disable micro-typographic setup inside <code>\add@accent</code> (reported by Stephan Hennig) . . . . .	68
<code>\MT@define@set@key@</code> : use comma lists instead of token lists . . . . .	72	<code>\MT@pdftex@no</code> : case 5: pdf $\TeX$ 1.30 . .	26
<code>\MT@find@file</code> : no longer wrap names in commands . . . . .	57	<code>\MT@permute@@@@@</code> : add ranges to the beginning of the lists . . . . .	89
<code>\MT@get@charwd</code> : warning for missing (resp. zero-width) characters . . .	44	<code>\MT@pr@split</code> : get character width once only . . . . .	43
<code>\MT@get@dimen@six</code> : new macro: test whether <code>\fontdimen6</code> is defined . . .	43	<code>\MT@scale</code> : bug fix: remove spaces in non-e- $\TeX$ variant (reported by Mark Rossi) . . . . .	38
<code>\MT@get@listname@</code> : made recursive . .	58	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>hyperref</code> is loaded . . . . .	31
<code>\MT@get@slot</code> : bug fix: expand active characters . . . . .	61	restore <code>csquotes</code> 's active characters . .	30
test whether <code>\(encoding)\(.)</code> is defined made more robust . . . . .	61	restore percent character if Spanish <code>babel</code> is loaded . . . . .	30
<code>\MT@get@unit</code> : new macro: get unit for codes . . . . .	47	<code>\MT@use@set</code> : bug fix: remove braces in first line . . . . .	76
<code>\MT@in@rlist</code> : made recursive . . . . .	37	<code>\MT@xadd</code> : simplified . . . . .	35
<code>\MT@is@active</code> : new macro: translate inputenc-defined characters . . . . .	63		
<code>\MT@is@letter</code> : warning for non-ASCII characters . . . . .	62		

### Version 1.9 (2005/10/28)

General: <code>\DeclareMicrotypeSet</code> : new key: font . . . . .	75	verbatim environment . . . . .	18
<code>\SetProtrusion</code> and <code>\SetExpansion</code> : new key: font . . . . .	82	add remark about Type 1 fonts required for automatic font expansion . . . . .	6
<code>\SetProtrusion</code> : value 'relative' renamed to 'character' for option unit . . . . .	84	Font Sets: add OT4 encoding to text sets . . . . .	106
allow context-specific font setup . .	67	add T5 encoding to text sets . . . .	106
disable expansion if both step and shrink are zero . . . . .	102	declare <code>qpl</code> and <code>qtm</code> (qfonts) as alias fonts of <code>ppl</code> resp. <code>ptm</code> . . . . .	107
disable microtype setup inside <code>hyperref</code> 's <code>\pdfstringdef</code> (reported by Hàn Thế Thành) . . . . .	31	Inheritance: add list for OT4 . . . . .	110
option unit: rename value relative to character . . . . .	97	add list for T5 (requested by Hàn Thế Thành) . . . . .	110
warning if user requested zero step . .	101	Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR . . . . .	115
Documentation: add hint about		settings for OT4 encoding (Computer Modern Roman, Palatino, Times) . . . . .	112

settings for T5 encoding (Computer Modern Roman) . . . . .	112	characters to the specified value before loading the lists . . . . .	47
\DisableLigatures: new macro: disable ligatures (requires pdf $\TeX$ 1.30) . . . . .	78	\MT@is@active: redone: use . . . . .	63
\microtypecontext: new macro: change setup context in the document . . . . .	69	\set@display@protect . . . . .	63
\MT@checklist@family: bug fix: add two missing \expandafters . . . . .	41	\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning . . . . .	62
\MT@define@option: fix: use true as the default value . . . . .	94	\MT@maybe@do: redone . . . . .	40
\MT@detokenize: fix the non-e- $\TeX$ version . . . . .	65	\MT@pdfTeX@no: compatibility with $\TeX$ Live hack (reported by Herbert Voß) . . . . .	26
\MT@exp@two@n: new macros: less \expandafters . . . . .	32	\MT@rem@from@clist: new macro: remove an item from a comma list . . . . .	36
\MT@get@opt: new key 'preset' to set all		\MT@scale@factor: generalized . . . . .	45
		\MT@toks: no longer use \toks@ . . . . .	28

### Version 1.9a (2005/12/05)

General: new option: defersetup, by default true . . . . .	96	\MT@get@highlevel: no longer check whether defaults have changed . . . . .	73
remove superfluous test whether \pickup@font has changed . . . . .	68	\MT@ifdefined@c@T: new macros: TRUE case only . . . . .	33
Documentation: add explanation for error message in DVI mode . . . . .	19	\MT@ifnumber: use \pdfmatch if available . . . . .	34
add explanation for error message with bitmap fonts . . . . .	19	\MT@ifstreq: use \pdfstrcmp if available . . . . .	35
Font Sets: declare mdbch (mathdesign) as an alias of Charter . . . . .	107	\MT@in@clist: fix . . . . .	36
Protrusion: settings for T5 encoded Charter . . . . .	112	\MT@info@missing@char: info instead of warning (after Michael Hoppe reported that the 'fl' ligature is missing in Palatino SC) . . . . .	45
\microtypesetup: inside the preamble, accepts all package options . . . . .	91	\MT@is@feature: new macro: check for pdf $\TeX$ feature . . . . .	71
\MT@check@font@cx: optimize context-sensitive setup . . . . .	68	\MT@map@clist@n: following $\LaTeX$ 3 . . . . .	35
\MT@define@set@key@: don't expand variables immediately (requested by Georg Verwey) . . . . .	72	\MT@permute@@@@: don't define permutations for unused encodings . . . . .	88
\MT@get@codes@name: 'file name/line number' as default list name . . . . .	81	\MT@rem@from@clist: fix . . . . .	36
		\MT@setup@microtype: defer setup until the end of the preamble . . . . .	31

### Version 2.0 ()

General: new option: babel, by default false (language-dependant setup suggested by Ulrich Drr) . . . . .	95	\MT@pdfTeX@no: case 6: pdf $\TeX$ 1.4x . . . . .	26
new option: letterspacing, by default 100 . . . . .	97	\SetExtraKerning: new macro: additional kerning . . . . .	80
\DeclareMicrotypeBabelHook: new command: interaction with babel . . . . .	78	\SetExtraSpacing: new macro: adjustment of interword spacing . . . . .	80
		\textls: new command: letterspacing . . . . .	56

## B Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

### Options:

DVIoutput	7
activate	5
auto	6
config	8
draft	7
expansion	5
factor	6
final	7
protrusion	5
selected	7
shrink	6
step	7
stretch	6
unit	6
verbose	8

### User Commands:

\DeclareCharacterInheritance	15
\DeclareMicrotypeAlias	15
\DeclareMicrotypeSet	8
\DeclareMicrotypeSet*	8
\DeclareMicrotypeSetDefault	10
\DisableLigatures	16
\LoadMicrotypeFile	16
\microtypecontext	17
\microtypesetup	8
\SetExpansion	13
\SetProtrusion	11
\UseMicrotypeSet	10
\@enc@update	609

### A

a0poster (package)	74
activate (option)	5, 94
\add@accent	1766, 1767
ae (package)	16, 107
ams* (package)	74
amssymb (package)	140
auto (option)	6, 95

### B

babel (option)	95, 157
babel (package)	2, 3, 22, 30, 58, 78, 104, 105, 107, 154, 156, 157

### C

\cf@encoding	609
chemsym (package)	30, 154
CJK (package)	18, 68
cm-super (package)	6
color (package)	7, 100
config (option)	8, 15, 21, 98, 155
contour (package)	100
crop (package)	100
csquotes (package)	20, 21, 30, 115, 156
\curr@fontshape	1327, 1332, 1737

### D

\DeclareCharacterInheritance	9, 15, 2552, 3191
\DeclareMicrotypeAlias	5, 15, 38, 2180, 3192
\DeclareMicrotypeBabelHook	16, 25, 2228
\DeclareMicrotypeSet	2, 8, 20, 21, 32, 1881, 3183
\DeclareMicrotypeSet*	8, 1881
\DeclareMicrotypeSetDefault	4, 10, 23, 2147
defersetup (option)	30, 91, 95, 96, 157
\define@newfont	1739
\DisableLigatures	8, 16, 24, 2213
\do@subst@correction	1736
docstrip (package)	24
draft (option)	7, 95, 96
dsfont (package)	43
DVIoutput (option)	7, 21, 95, 96, 154

### E

eco (package)	16, 107
\efcode	1021, 1055, 1056, 1108, 1111
eucal (package)	145
eufrak (package)	144
expansion (option)	5, 10, 94, 100

### F

\f@encoding	609
\f@size	1327, 1332, 1737
factor (option)	6, 12, 22, 97, 154
fancyvrb (package)	18
final (option)	7, 22, 95, 96, 153

- fix-cm (package) ..... 14
- \font ..... 270
- \font@name 1739, 1751, 1753, 1774, 1794
- \fontcharwd ..... 817
- \fontdimen .... 759, 766, 837, 940, 943
- fontinst (package) ..... 148
- fontinstallationguide (package) .. 129
- french (package) ..... 155
- frenchpro (package) ..... 22, 64, 154
- G**
- german (package) ..... 22, 27, 153
- graphics (package) ..... 7, 100
- H**
- hfoldsty (package) ..... 16, 107
- hyperref (package) .. 7, 31, 66, 100, 156
- I**
- IEEEtran (package) ..... 67
- \iffontchar ..... 841
- \ifMT@auto ..... 139, 3257, 3304
- \ifMT@babel ..... 139, 3347, 3400
- \ifMT@do ..... 628, 712, 1375
- \ifMT@document ..... 150, 2684
- \ifMT@draft ..... 139, 3171
- \ifMT@expansion ..... 139, 3221, 3301
- \ifMT@inlist@ 479, 534, 657, 679, 696,  
705, 725, 1329, 1408, 1419,  
1507, 1758, 1782, 1846, 1853, 2198
- \ifMT@kerning .....  
.... 139, 1323, 3362, 3389, 3425
- \ifMT@noligatures ..... 139
- \ifMT@nonselected ... 1002, 1520, 1529
- \ifMT@noreset ..... 1583, 1590
- \ifMT@opt@auto ..... 2922, 3260, 3272
- \ifMT@opt@DVI ..... 2922, 3203
- \ifMT@opt@expansion ..... 2922, 3198
- \ifMT@protrusion ..... 139, 191, 3204
- \ifMT@selected ..... 139, 3287, 3307
- \ifMT@spacing ..... 139, 3336, 3381
- ifpdf (package) ..... 7, 100
- inputenc (package) .. 18, 61, 63–65, 156
- J**
- jurabib (package) ..... 66
- K**
- kerning (option) ..... 95
- keyval (package) ..... 28, 43, 72, 108
- \knaccode . 1272, 1273, 1285, 1289, 1296
- \knbccode . 1265, 1266, 1284, 1288, 1295
- \knbscode . 1158, 1159, 1184, 1189, 1197
- L**
- ledmac (package) ... 20, 21, 29, 66, 156
- ledpar (package) ..... 20
- \leftmargin kern ..... 200
- letterspacing (option) ..... 97, 157
- lineno (package) ..... 29
- listings (package) ..... 18
- lmodern (package) ..... 6, 117
- \LoadMicrotypeFile .....  
..... 10, 16, 41, 2194, 3193
- \lpcode ..... 752, 790, 791, 949, 953
- \lststyle ..... 18, 264, 1321, 1359
- M**
- mathdesign (package) ..... 16, 107, 157
- memoir (package) ..... 67
- \Microtype@Hook ..... 98, 3144
- \microtypecontext .... 12, 17, 1343,  
1816, 3195, 3408, 3411, 3419, 3422
- \microtypesetup ..... 8, 11, 2811, 3194
- minimal (package) ..... 32
- \MT@abbr@ex ..... 580
- \MT@abbr@ex@c ..... 580
- \MT@abbr@ex@inh ..... 580
- \MT@abbr@kn ..... 580
- \MT@abbr@kn@c ..... 580
- \MT@abbr@kn@inh ..... 580
- \MT@abbr@nl ..... 580, 1364
- \MT@abbr@pr ..... 580
- \MT@abbr@pr@c ..... 580
- \MT@abbr@pr@inh ..... 580
- \MT@abbr@sp ..... 580
- \MT@abbr@sp@c ..... 580
- \MT@abbr@sp@inh ..... 580
- \MT@active@features ... 1776, 1779,  
1791, 1798, 1833, 1845, 2215,  
3205, 3302, 3337, 3361, 3443, 3444
- \MT@addto@setup .... 232, 239, 1735,  
1816, 1817, 1957, 2068, 2812,  
3033, 3170, 3335, 3380, 3399, 3442
- \MT@allowed@features ..... 1916
- \MT@auto ..... 1011,  
1104, 3256, 3269, 3280, 3284
- \MT@auto@ ..... 999, 1011, 1015, 1073
- \MT@autofalse ..... 138, 3267, 3279
- \MT@autotruer ..... 138, 3113
- \MT@babelfalse ..... 138, 3180
- \MT@babeltrue ..... 138
- \MT@begin@catcodes .....  
1409, 1410, 1433, 2202, 2250,  
2274, 2296, 2317, 2554, 3126, 3127
- \MT@catcodes ..... 115, 135

- \MT@char ..... 780, 790, 791,  
796, 797, 800, 802, 817, 822,  
841, 949, 950, 952–954, 1043,  
1055, 1056, 1058, 1060, 1108,  
1110, 1111, 1147, 1158, 1159,  
1165, 1166, 1172, 1173, 1176,  
1178, 1184–1186, 1188–1191,  
1254, 1265, 1266, 1272, 1273,  
1276, 1278, 1284, 1285, 1287–  
1289, 1561, 1566, 1576, 1579,  
1580, 1585, 2630, 2631, 2636, 2639
- \Mt@char ..... 1561, 1568, 1575,  
1579, 1596, 1598, 1604, 1605,  
1607, 1620, 1621, 1624, 1625,  
1628, 1629, 1633, 1635, 1667, 1712
- \MT@charstring ..... 1576, 1666, 1671
- \MT@check@font ..... 1757, 1773, 1836
- \MT@check@font@cx ..... 1777, 1836
- \MT@check@range ..... 2773, 2775
- \MT@check@range@ ..... 2776, 2778
- \MT@check@rlist ..... 2714, 2764
- \MT@check@rlist@ ..... 2765, 2767
- \MT@checklist@ ..... 635, 648, 1369
- \MT@checklist@family ..... 689
- \MT@checklist@font ..... 668
- \MT@checklist@size ..... 718
- \MT@checksetup .....  
... 2822, 2833, 2840, 2861, 2883
- \MT@clist@break . 448, 663, 681, 713, 731
- \MT@clist@function ..... 448
- \MT@cnt@encoding .... 2651, 2658, 2659
- \MT@cnt@family ..... 2656, 2665, 2666
- \MT@cnt@series ..... 2663, 2672, 2673
- \MT@cnt@shape ..... 2670, 2678, 2679
- \MT@config@file ..... 3116,  
3124, 3125, 3129, 3130, 3134
- \MT@context ... 1483, 1495, 1514, 1540
- \MT@count ..... 387,  
388, 551, 809, 817, 818, 823,  
824, 829, 830, 834, 837, 874, 934
- \MT@curr@file 1413, 1414, 1421, 1422,  
2204, 2205, 2331, 2589, 3129, 3449
- \MT@curr@list@name ..... 744, 877,  
996, 1069, 1132, 1239, 1395,  
1553, 1710, 1715, 1720, 1726, 1731
- \MT@curr@set@name 1925, 1927–1929,  
1931, 1932, 1945, 1950, 1955,  
1958, 1984, 1988, 2019, 2050,  
2056, 2061, 2066, 2069, 2333, 2335
- \MT@declare@char@inh ..... 2559,  
2560, 2562, 2563, 2570, 2578
- \MT@declare@codes .....  
... 2436, 2469, 2470, 2472, 2473
- \MT@declare@sets ..... 1898,  
1899, 1901, 1902, 1909, 1926, 2217
- \MT@DeclareSet ..... 1885, 1886, 1888
- \MT@DeclareSet@ ..... 1889, 1892, 1895
- \MT@DeclareSetAndUseIt 1883, 1884, 1888
- \MT@def@bool@opt .....  
... 2999, 3015, 3019, 3028, 3037
- \MT@def@n ..... 282, 2232, 2254,  
2278, 2300, 2321, 2439, 2526, 2582
- \MT@def@num@opt ..... 3063, 3078
- \MT@def@simple@bool@opt ... 3015, 3018
- \MT@default@ex@set .. 2168, 3139, 3311
- \MT@default@kn@set .. 2168, 3142, 3369
- \MT@default@pr@set .. 2168, 3138, 3214
- \MT@default@sp@set .. 2168, 3141, 3343
- \MT@define@code@key .....  
... 2376, 2452–2455, 2608–2610
- \MT@define@code@key@font .....  
..... 2400, 2457, 2612
- \MT@define@code@key@size .....  
..... 2387, 2456, 2611
- \MT@define@context .....  
... 1841, 1862, 1863, 1865, 1866
- \MT@define@ex@ck@key . 2522, 2535–2537
- \MT@define@inh@key@encoding 2591, 2607
- \MT@define@inh@keys ..... 2606
- \MT@define@key@unit . 2494, 2519, 2520
- \MT@define@option ... 2925, 2961, 2962
- \MT@define@option@ .. 2968, 2996, 2997
- \MT@define@optionX .. 2811, 2871, 2872
- \MT@define@optionX@ . 2874, 2904, 2905
- \MT@define@set@key@ . 1936–1939, 1943
- \MT@define@set@key@font ... 1941, 2054
- \MT@define@set@key@size ... 1940, 1977
- \MT@define@set@keys ..... 1935,  
2105, 2106, 2108, 2109, 2111
- \MT@detokenize ..... 1570, 1704
- \MT@dimen@six ..... 758, 810, 834, 874
- \MT@do@font . 545, 751, 1021, 1196, 1294
- \MT@documentfalse ..... 149
- \MT@documenttrue ..... 149, 3446
- \MT@dofalse .....  
627, 639, 662, 684, 701, 730, 1373
- \MT@dotrue .....  
627, 631, 659, 698, 707, 727, 1365
- \MT@draftfalse ..... 138, 3039
- \MT@drafttrue ..... 138, 3041
- \MT@edef@n ..... 283,  
1519, 1544, 1857, 2136, 2140,  
2172, 2177, 2188, 2331, 2382,



- 2405, 2588, 2596, 2736, 2744,  
2945, 2947, 2982, 2984, 3069, 3084
- \MT@encoding . . . . . 618, 1478, 1490,  
1570, 1662, 1697, 1719, 1725, 1731
- \MT@encoding@check . . . . . 2594, 2599
- \MT@end@catcodes . 1411, 1450, 2210,  
2255, 2279, 2301, 2322, 2576, 3128
- \MT@error . . . . . 43, 182, 1918
- \MT@ex@c@name . . . . . 995–997,  
1074, 1075, 1080, 1081, 1086,  
1087, 1092, 1093, 1098, 1099,  
1115, 1521, 2257, 2278, 2347, 2351
- \MT@ex@context . . . . . 1827, 1868
- \MT@ex@do . . . . . 998, 1033
- \MT@ex@doc@contexts . . . . . 1868
- \MT@ex@factor . . . . . 151, 1012, 1078
- \MT@ex@factor@ . . . . . 1012,  
1023, 1026, 1045, 1046, 1073
- \MT@ex@inh@name . . . . . 1057, 1058, 1060
- \MT@ex@level . . . . . 151, 3303, 3305
- \MT@ex@max . . . . . 166, 1048, 1049
- \MT@ex@min . . . . . 166, 1051, 1052
- \MT@ex@setname . . . . .  
... 2133, 3308, 3309, 3311, 3312
- \MT@ex@split . . . . . 1035, 1039
- \MT@exp@one@n . . . . . 288, 454, 606, 608,  
673, 1418, 1773, 1780, 1812,  
1845, 1851, 1909, 1957, 2068,  
2126, 2161, 2197, 2570, 3408, 3419
- \MT@exp@string . . . . . 287,  
603, 761, 843, 850, 1332, 1533
- \MT@exp@two@c . . . . . 289, 341,  
349, 605, 1569, 1576, 1665, 1708
- \MT@exp@two@n . . . . .  
. 290, 694, 703, 1326, 1949, 2060
- \MT@expansion . . . . . 611, 989, 3328
- \MT@expansionfalse . . . . .  
. . . . . 138, 3176, 3199, 3297
- \MT@expansiontrue . . . . . 138, 3112
- \MT@extra@auto . . . . . 2257, 2542, 2548
- \MT@extra@context . . . . . 1868,  
2236, 2257, 2282, 2303, 2409,  
2410, 2465, 2711, 2717, 2721,  
2725, 2728, 2732, 2733, 2736, 2772
- \MT@extra@factor . . 2236, 2257, 2282,  
2303, 2344–2346, 2349, 2351, 2449
- \MT@extra@preset . . . . .  
... 2236, 2257, 2282, 2303, 2460
- \MT@extra@shrink . . . . . 2257
- \MT@extra@step . . . . . 2257
- \MT@extra@stretch . . . . . 2257
- \MT@extra@unit . . . . . 2236, 2282,  
2303, 2476, 2486, 2497, 2506, 2509
- \MT@factor@default . . . . .  
. . . . . 176, 3081, 3087, 3208
- \MT@family . . . . .  
. 606, 618, 695, 1479, 2189, 2190
- \MT@familyalias . 607, 608, 624, 702,  
704, 706, 708, 1489, 1491, 2191
- \MT@feat . . . 628, 763, 855, 857, 859,  
860, 863, 864, 870, 872, 875,  
882–886, 888, 890–894, 897,  
898, 903, 907, 910, 913, 916,  
917, 926, 927, 932, 1385, 1387,  
1390, 1391, 1395–1397, 1399,  
1515, 1517, 1519, 1523, 1524,  
1528, 1532, 1535, 1542, 1544,  
1546, 1547, 1549, 1552, 1555, 1558
- \MT@file@list . . . . . 1405, 1407,  
1415, 1418, 1423, 1426, 2197, 2201
- \MT@find@file . . . . . 606, 608, 1405
- \MT@fix@font@spec . . . 1954, 1973, 2065
- \MT@font . . . . . 603, 605, 677,  
680, 683, 752, 753, 759, 761,  
766, 790, 791, 796, 797, 817,  
822, 837, 841, 843, 850, 940,  
943, 949, 950, 953, 954, 999,  
1015, 1021, 1055, 1056, 1108,  
1111, 1158, 1159, 1165, 1166,  
1172, 1173, 1184–1186, 1189–  
1191, 1197–1199, 1265, 1266,  
1272, 1273, 1284, 1285, 1288,  
1289, 1295, 1296, 1376, 1462,  
1533, 1734, 1750, 1751, 1753,  
1763, 1773, 1775, 1780, 1802, 1813
- \MT@font@list . . . 1733, 1773–1775, 1834
- \MT@get@axis . . . . .  
... 2076–2079, 2092, 2422–2425
- \MT@get@basefamily . . . . . 1417, 1451
- \MT@get@char@unit . . . . .  
. . . . . 781, 905, 930, 1148, 1255
- \MT@get@charwd . . . . . 815, 834, 908, 930
- \MT@get@codes@name . . . . .  
... 2244, 2268, 2290, 2311, 2325
- \MT@get@config . . . . . 3116
- \MT@get@dimen@six 740, 758, 1128, 1235
- \MT@get@ex@opt . . . . . 992, 1006, 1073
- \MT@get@font . . . . . 2059, 2072
- \MT@get@font@and@size . . . . . 2404, 2418
- \MT@get@font@dimen . . . . . 836, 911
- \MT@get@highlevel . . . . .  
... 1948, 1961, 2094, 2381, 2595

- \MT@get@inh@list ..... 742, 994, 1130, 1237, 1540  
 \MT@get@inh@name ..... 2580, 2587  
 \MT@get@listname .... 1461, 1517, 1542  
 \MT@get@listname@ ..... 1461  
 \MT@get@opt ..... 741, 881, 1129, 1236  
 \MT@get@range ..... 1981, 1991, 2391  
 \MT@get@size ..... 1998,  
     2003, 2014, 2033, 2086, 2433  
 \MT@get@slot ..... 779,  
     1042, 1146, 1253, 1561, 2629, 2635  
 \MT@get@space@unit . 905, 979, 1156,  
     1163, 1170, 1217, 1221, 1263, 1270  
 \MT@get@unit ..... 913, 920  
 \MT@get@unit@ ..... 920  
 \MT@gobble@settings ..... 758  
 \MT@gobble@three ..... 2912, 2916  
 \MT@if@list@exists .....  
     739, 991, 1005, 1127, 1234, 1514  
 \MT@ifdefined@c@T ..... 221, 245,  
     294, 799, 1057, 1175, 1275,  
     2189, 2334, 2344, 2700, 3144, 3148  
 \MT@ifdefined@c@TF . 193, 294, 1340,  
     1518, 1543, 2710, 3138, 3139,  
     3141, 3142, 3211, 3308, 3340, 3366  
 \MT@ifdefined@n@T .....  
     . 294, 650, 670, 691, 720, 800,  
     916, 1058, 1176, 1276, 1497,  
     1928, 2184, 2327, 2372, 2659,  
     2666, 2673, 2679, 2711, 2725, 3154  
 \MT@ifdefined@n@TF .. 294, 624, 633,  
     651, 671, 692, 721, 882, 890,  
     1074, 1080, 1086, 1092, 1098,  
     1367, 1390, 1485, 1570, 1917,  
     2135, 2139, 2170, 2326, 2686,  
     2743, 2759, 2944, 2981, 3406, 3417  
 \MT@ifdimen 385, 2045, 2485, 2508, 3101  
 \MT@ifdocument ..... 150  
 \MT@ifempty 357, 788, 794, 1155, 1162,  
     1169, 1262, 1269, 1354, 1847,  
     1897, 1907, 1963, 1969, 1992,  
     1993, 2006, 2007, 2073, 2080,  
     2095, 2114, 2124, 2149, 2159,  
     2419, 2427, 2438, 2443, 2448,  
     2459, 2464, 2524, 2558, 2568,  
     2600, 2819, 2880, 2930, 2973,  
     3001, 3046, 3065, 3080, 3091, 3118  
 \MT@ifeq ..... 394,  
     523, 524, 2025, 2779–2781, 2791  
 \MT@ifgt .. 394, 2016, 2783, 2784, 2797  
 \MT@iflt 394, 528, 529, 2792, 2793, 2796  
 \MT@ifnumber 368, 1627, 2525, 3068, 3083  
 \MT@ifstreq ..... 415, 1312,  
     1386, 1456–1458, 1516, 1592,  
     1811, 2478, 2484, 2499, 2505,  
     2507, 2821, 2828, 2832, 2839,  
     2882, 2888, 2933, 2935, 2938,  
     2941, 2976, 2978, 3004, 3005,  
     3049, 3050, 3055, 3094, 3100, 3348  
 \MT@in@clist .....  
     479, 654, 674, 694, 703, 1326,  
     1407, 1418, 1773, 1780, 1845, 2197  
 \MT@in@rlist ..... 515, 723, 1506  
 \MT@in@rlist@ ..... 515  
 \MT@in@rlist@@ ..... 515  
 \MT@in@tlist ..... 503, 1851  
 \MT@in@tlist@ ..... 503  
 \MT@increment .....  
     .... 551, 2658, 2665, 2672, 2678  
 \MT@info .....  
     43, 57, 603, 2137, 2824, 2830,  
     2836, 2843, 2885, 2890, 2915, 2919  
 \Mt@info ..... 1671  
 \MT@info@missing@char ... 818, 824, 839  
 \MT@info@nl ..... 43, 54,  
     58, 194, 602, 841, 848, 3045,  
     3125, 3202, 3207, 3212, 3215,  
     3219, 3304, 3309, 3312, 3329,  
     3339, 3341, 3344, 3359, 3365,  
     3367, 3370, 3373, 3385, 3393, 3402  
 \MT@inh@do ..... 1554, 2620  
 \MT@inh@name .. 1552, 1553, 2638, 2645  
 \MT@inh@split ..... 2622, 2626  
 \MT@inlist@false .....  
     ..... 478, 483, 504, 516, 1788  
 \MT@inlist@true .....  
     ... 478, 485, 511, 525, 530, 1788  
 \MT@is@active ..... 1569, 1640  
 \MT@is@char ..... 1576, 1665, 1671  
 \MT@is@composite ..... 1573, 1693  
 \MT@is@feature .....  
     ... 1908, 1916, 2125, 2160, 2569  
 \MT@is@letter .. 1567, 1594, 1668, 1701  
 \MT@is@number ..... 1610, 1615  
 \MT@is@symbol ..... 1571, 1659  
 \MT@iterate ..... 539  
 \MT@kerning ..... 614, 1227, 3396  
 \MT@kerningfalse .... 138, 3179, 3395  
 \MT@kerningtrue ..... 138, 1336  
 \MT@kn@c@name .....  
     ... 1238–1240, 1303, 2303, 2321  
 \MT@kn@context ..... 1312, 1828, 1868  
 \MT@kn@do ..... 1241, 1244  
 \MT@kn@doc@contexts ..... 1868

- `\MT@kn@factor` ..... 151  
`\MT@kn@factor@` ..... 882, 1313, 1315  
`\MT@kn@inh@name` ..... 1275, 1276, 1278  
`\MT@kn@max` ..... 166  
`\MT@kn@min` ..... 166  
`\MT@kn@setname` ..... 1328,  
     1331, 1339, 3366, 3367, 3369, 3370  
`\MT@kn@split` ..... 1246, 1250  
`\MT@kn@split@val` ..... 1256, 1260  
`\MT@kn@unit` ..... 151  
`\MT@kn@unit@` ..... 890, 1306  
`\MT@led@kern` ..... 190  
`\MT@led@unhbox@line` ..... 190  
`\MT@led@mac@setup` ..... 190, 241  
`\MT@let@cn` . 285, 286, 625, 745, 997,  
     1075, 1081, 1087, 1093, 1099,  
     1133, 1240, 1385, 1396, 1511,  
     1515, 1549, 1552, 2037, 2333, 3066  
`\MT@let@nc` ..... 284, 1528,  
     1555, 1558, 1783, 1834, 1931,  
     1945, 2056, 2335, 2351, 2579,  
     2745, 2757, 2835, 2842, 2939, 2942  
`\MT@let@nn` ..... 286,  
     883, 888, 891, 903, 1850, 2373  
`\MT@letterspacing` 151, 1341, 3375, 3376  
`\MT@letterspacing@` .....  
     ..... 1314, 1340, 1341, 1353  
`\MT@letterspacing@default` .. 180, 3376  
`\MT@listname` ..... 1461,  
     1508, 1511, 1518, 1519, 1543, 1544  
`\MT@load` ..... 2236,  
     2257, 2282, 2303, 2334, 2335, 2444  
`\MT@load@list` 743, 995, 1131, 1238, 1383  
`\MT@loop` ..... 539, 547, 2756  
`\MT@lower` ..... 1985, 1991, 2395  
`\MT@make@string` . 579, 1975, 2183, 2196  
`\MT@map@clist@` ..... 448  
`\MT@map@clist@c` .....  
     ..... 448, 1779, 1791, 1798,  
     1833, 1905, 2122, 2157, 2566, 3152  
`\MT@map@clist@n` 448, 632, 1366, 1946,  
     1979, 2057, 2230, 2379, 2389,  
     2402, 2632, 2816, 2877, 2928, 2971  
`\MT@map@tlist@` ..... 464  
`\MT@map@tlist@c` .....  
     .... 464, 506, 517, 801, 1059,  
     1177, 1277, 1465, 1804, 2712, 2771  
`\MT@map@tlist@n` . 464, 1445, 3016, 3076  
`\MT@maybe@do` . 628, 736, 989, 1121, 1228  
`\MT@MT` ..... 42,  
     43, 44, 47, 49, 50, 108, 183, 185,  
     1647, 2915, 2919, 3119, 3203, 3352  
`\MT@next@listname` ..... 1468, 1477  
`\MT@next@listname@` .. 1486, 1498, 1504  
`\MT@nl@context` ..... 1870  
`\MT@nl@doc@contexts` ..... 1873  
`\MT@nl@setname` ..... 2218  
`\MT@noligatures` ..... 616, 1362  
`\MT@noligaturesfalse` ..... 138  
`\MT@noligaturestrue` ..... 138, 2216  
`\MT@nonselectedfalse` ..... 1001, 1016  
`\MT@nonselectedtrue` ..... 1001, 1004  
`\MT@noestfalse` .....  
     ... 1589, 1592, 1600, 1609, 1630  
`\MT@noesttrue` ..... 1564, 1589, 1618  
`\MT@old@cmd` ..... 26, 31, 34, 37, 40  
`\MT@opt@autofalse` ..... 2921  
`\MT@opt@autottrue` ..... 2921  
`\MT@opt@DVIfalse` ..... 2921  
`\MT@opt@DVIttrue` ..... 2921, 3021, 3024  
`\MT@opt@expansionfalse` ..... 2921  
`\MT@opt@expansiontrue` ..... 2921  
`\MT@orig@add@accent` ..... 1766  
`\MT@orig@foreign@language` . 3414, 3416  
`\MT@orig@pickupfont` .. 262, 1739, 1768  
`\MT@orig@select@language` .. 3403, 3405  
`\MT@pdfcprot@error` ..... 181, 240  
`\MT@pdftex@no` .. 60, 94, 100, 104, 3283  
`\MT@permute` ..... 2249,  
     2273, 2295, 2316, 2585, 2650  
`\MT@permute@` ..... 2650  
`\MT@permute@@` ..... 2650  
`\MT@permute@@@` ..... 2650  
`\MT@permute@@@@` ..... 2650  
`\MT@permute@@@@@` ..... 2677, 2682  
`\MT@permute@@@@@@` ..... 2690, 2692  
`\MT@permute@define` .....  
     ..... 2683, 2693–2695, 2741  
`\MT@permute@reset` ..... 2653, 2747  
`\MT@permute@reset@` .. 2748–2751, 2754  
`\MT@permute@list` .....  
     2247, 2249, 2271, 2293, 2314,  
     2405, 2406, 2413, 2583, 2706,  
     2707, 2711, 2717, 2721, 2725–  
     2728, 2736, 2737, 2772, 2805, 2806  
`\MT@pickupfont` ..... 1765, 1770  
`\MT@pr@c@name` 743–745, 959, 2236, 2254  
`\MT@pr@context` ..... 1827, 1868  
`\MT@pr@do` ..... 746, 770  
`\MT@pr@doc@contexts` ..... 1868  
`\MT@pr@factor` ..... 151, 3208, 3209  
`\MT@pr@factor@` ..... 882  
`\MT@pr@inh@name` ..... 799, 800, 802  
`\MT@pr@level` ..... 151, 3206, 3207

- \MT@pr@max ..... 166
- \MT@pr@min ..... 166
- \MT@pr@setname .....
  - ... 2133, 3211, 3212, 3214, 3215
- \MT@pr@split ..... 772, 776
- \MT@pr@split@val ..... 782, 786
- \MT@pr@unit@ ..... 151, 3102, 3210
- \MT@pr@unit@ ..... 890, 962
- \MT@preset@aux ..... 964, 966, 968,
  - 969, 972, 1308, 1310, 1317, 1318
- \MT@preset@aux@factor .....
  - ... 964, 972, 1211–1213, 1308
- \MT@preset@aux@space .....
  - ... 966, 972, 1215, 1310
- \MT@preset@ex ..... 1114
- \MT@preset@kn ..... 1301
- \MT@preset@kn@ ..... 1301
- \MT@preset@pr ..... 957
- \MT@preset@pr@ ..... 957
- \MT@preset@sp ..... 1204
- \MT@preset@sp@ ..... 1204
- \MT@ProcessOptionsWithKV .. 3149, 3169
- \MT@protrusion ..... 610, 736, 3218
- \MT@protrusionfalse ..... 138, 3175
- \MT@protrusiontrue ..... 138, 3109
- \MT@rbba@expansion ..... 595
- \MT@rbba@kerning ..... 595
- \MT@rbba@protrusion ..... 595
- \MT@rbba@spacing ..... 595
- \MT@register@font ... 1760, 1775, 1837
- \MT@register@font@cx ..... 1797, 1837
- \MT@register@subst@font .....
  - ... 1754, 1774, 1838
- \MT@register@subst@font@cx 1790, 1838
- \MT@rem@from@clist ..... 490, 1812
- \MT@rem@from@lists ..... 1806, 1810
- \MT@repeat ..... 539, 549, 2762
- \MT@requires@etex .....
  - 271, 294, 552, 563, 815, 839, 1704
- \MT@requires@pdftex .....
  - ... 93, 192, 368, 415, 807,
  - 1024, 1120, 1227, 1321, 1362,
  - 2213, 2541, 2547, 3111, 3258, 3334
- \MT@reset@context ..... 1822, 1825
- \MT@reset@ef@codes ... 993, 1014, 1024
- \MT@reset@ef@codes@ . 1019, 1027, 1031
- \MT@reset@kn@codes ..... 1233, 1292
- \MT@reset@kn@codes@ ..... 1292
- \MT@reset@pr@codes ..... 738, 756
- \MT@reset@pr@codes@ ..... 756
- \MT@reset@sp@codes ..... 1126, 1194
- \MT@reset@sp@codes@ ..... 1194
- \MT@restore@catcodes ..... 126, 136
- \MT@saved@setupfont ..... 2913, 2920
- \MT@scale ..... 563,
  - 810, 856, 871, 874, 1046, 1314
- \MT@scale@factor .....
  - ... 812, 831, 854, 974, 1116
- \MT@scale@to@em .....
  - 789, 795, 807, 828, 980, 1157,
  - 1164, 1171, 1218, 1222, 1264, 1271
- \MT@selectedfalse ..... 138
- \MT@selectedtrue ..... 138
- \MT@series ..... 618, 1480, 1492
- \MT@set@all@ex ..... 1019, 1117
- \MT@set@all@kn ..... 1292, 1319
- \MT@set@all@pr ..... 749, 756, 970
- \MT@set@all@sp ..... 1194, 1225
- \MT@set@default@set ..... 2150,
  - 2151, 2153, 2154, 2161, 2168
- \MT@set@ex@codes ... 1018, 3288, 3290
- \MT@set@ex@codes@n .. 1002, 1018, 3290
- \MT@set@ex@codes@s ..... 990, 3288
- \MT@set@ex@heirs ..... 1061, 1107
- \MT@set@ex@list ..... 2275, 2277
- \MT@set@ex@opt ..... 2269, 2343
- \MT@set@inh@list ..... 2555, 2557
- \MT@set@kn@codes ..... 1232
- \MT@set@kn@heirs ..... 1279, 1283
- \MT@set@kn@list ..... 2318, 2320
- \MT@set@kn@opt ..... 2312, 2365
- \MT@set@opt@ ... 2339–2341, 2353–
  - 2357, 2361–2363, 2366–2368, 2371
- \MT@set@pr@codes ..... 737
- \MT@set@pr@heirs ..... 803, 948
- \MT@set@pr@list ..... 2251, 2253
- \MT@set@pr@opt ..... 2245, 2338
- \MT@set@sp@codes ..... 1125
- \MT@set@sp@heirs ..... 1179, 1183
- \MT@set@sp@list ..... 2297, 2299
- \MT@set@sp@opt ..... 2291, 2360
- \MT@setup@contexts ..... 1819, 1832
- \MT@setup@microtype .....
  - .. 231, 234, 235, 238, 3031, 3032
- \MT@setupfont .....
  - 601, 1759, 2913, 2916, 2920, 3182
- \MT@setupfont@hook .....
  - ... 219, 242, 244, 249, 257, 604
- \MT@shape ..... 618, 1481, 1493
- \MT@shrink ..... 151,
  - 1009, 1090, 3225, 3227, 3229,
  - 3233, 3234, 3237, 3241, 3293, 3306
- \MT@shrink@ ... 999, 1009, 1015, 1073
- \MT@shrink@default ..... 176, 3229

- \MT@size .. 524, 528, 529, 618, 726, 729  
 \MT@size@name ..... 515, 1508  
 \MT@sp@cc@name .....  
     ... 1131–1133, 1206, 2282, 2300  
 \MT@sp@context ..... 1828, 1868, 3348  
 \MT@sp@do ..... 1134, 1137  
 \MT@sp@doc@contexts ..... 1868  
 \MT@sp@factor ..... 151  
 \MT@sp@factor@ ..... 882  
 \MT@sp@inh@name ..... 1175, 1176, 1178  
 \MT@sp@max ..... 166  
 \MT@sp@min ..... 166  
 \MT@sp@setname 3340, 3341, 3343, 3344  
 \MT@sp@split ..... 1139, 1143  
 \MT@sp@split@val ..... 1149, 1153  
 \MT@sp@unit ..... 151  
 \MT@sp@unit@ ..... 890, 1209  
 \MT@spacing .... 613, 1120, 3358, 3388  
 \MT@spacingfalse ..... 138, 3178, 3387  
 \MT@spacingtrue ..... 138  
 \MT@split@name ..... 605, 618  
 \MT@split@val ..... 1153  
 \MT@step ..... 151,  
     1010, 1096, 3232, 3248, 3255, 3307  
 \MT@step@ ..... 999, 1010, 1015, 1073  
 \MT@step@default ..... 176  
 \MT@stretch ..... 151, 1008,  
     1084, 3222, 3223, 3226, 3227,  
     3233, 3235, 3240, 3243, 3292, 3306  
 \MT@stretch@ ... 999, 1008, 1015, 1073  
 \MT@stretch@default ..... 176, 3223  
 \MT@temp ..... 2075, 2103, 2421  
 \MT@tempencoding 2685, 2686, 2696, 2703  
 \MT@tempfamily ..... 2697  
 \MT@tempseries ..... 2698  
 \MT@tempshape ..... 2699  
 \MT@tempsize .....  
     ... 2700, 2710, 2713, 2718, 2752  
 \MT@test@ast ..... 1962, 1967  
 \MT@testrest ..... 1590  
 \MT@tlist@break .....  
     ... 464, 512, 536, 1470, 2808  
 \MT@toks ..... 138, 234,  
     235, 841, 849, 877, 1069, 1565,  
     1598, 1605, 1651, 1712, 1724, 1730  
 \MT@try@order ..... 1465, 1473  
 \MT@upper ..... 1985, 1991, 2395  
 \MT@use@set ..... 2115,  
     2116, 2118, 2119, 2126, 2133  
 \MT@val ..... 1847–  
     1849, 1852, 1854, 1857, 1947,  
     1951, 1962, 1964, 1980–1982,  
     1994, 1997, 1999, 2002, 2004,  
     2005, 2013, 2015, 2016, 2018,  
     2021, 2023, 2034, 2037, 2041,  
     2045–2047, 2049, 2051, 2058,  
     2059, 2062, 2082, 2084, 2087–  
     2089, 2093, 2095, 2097, 2099,  
     2380, 2382, 2390–2392, 2403,  
     2404, 2408, 2414, 2429, 2431,  
     2593, 2594, 2596, 2601, 2631,  
     2638, 2644, 2645, 2817, 2819,  
     2821, 2828, 2832, 2839, 2847,  
     2878, 2880, 2882, 2888, 2893,  
     2929, 2930, 2933, 2935, 2938,  
     2941, 2944, 2945, 2950, 2972,  
     2973, 2976, 2978, 2981, 2982, 2987  
 \MT@vinfo ..... 43, 54,  
     602, 645, 885, 894, 898, 932,  
     1076, 1082, 1088, 1094, 1101,  
     1377, 1391, 1414, 1422, 1425,  
     1521, 1523, 1530, 1826, 1849,  
     2199, 2205, 3045, 3054, 3407, 3418  
 \Mt@warn@ascii ..... 1607, 1711  
 \MT@warn@axis@empty .....  
     ... 2081, 2096, 2101, 2428  
 \MT@warn@code@too@large . 861, 865, 869  
 \MT@warn@err ..... 43, 3051, 3052  
 \MT@warn@ex@too@large 1049, 1052, 1067  
 \MT@warn@number@too@large . 1634, 1717  
 \MT@warn@preset@towidth .....  
     ... 963, 983, 1210, 1307  
 \MT@warn@unknown ..... 1581, 1728  
 \MT@warn@unknown@i ..... 1584, 1722  
 \MT@warning 26, 43, 1347, 1387, 1399,  
     1532, 1929, 2017, 2049, 2102,  
     2142, 2174, 2185, 2207, 2222,  
     2328, 2479, 2488, 2500, 2511,  
     2528, 2544, 2602, 2704, 2726,  
     2805, 2846, 2865, 2892, 3051, 3095  
 \MT@warning@nl .. 43, 103, 206, 211,  
     251, 760, 875, 925, 984, 1068,  
     1712, 1718, 1723, 1729, 1741,  
     2346, 2949, 2986, 3006, 3052,  
     3056, 3070, 3085, 3104, 3133,  
     3145, 3172, 3250, 3261, 3273,  
     3294, 3349, 3382, 3390, 3432, 3436  
 \MT@xadd ..... 434,  
     1330, 1415, 1423, 1426, 1792,  
     1800, 1854, 1983, 2201, 2393, 2637  
 \MT@xaddb ..... 441, 2412, 2716
- N
- ngerman (package) ..... 22

- `\normalfont` ..... 270, 3316, 3324  
`\nullfont` ..... 270
- P**
- `\pdfadjustinterwordglue` 69, 2904, 3338  
`\pdfadjustspacing` ... 2872, 3303, 3321  
`\pdfappendkern` ..... 1337, 2906, 3364  
`pdfcpot` (package) .....  
     ..... 4, 18, 19, 29, 44, 66, 153  
`\pdffontexpand` ..... 999, 1015  
`\pdfmatch` ..... 370  
`\pdfnoligatures` ..... 1376  
`\pdfoutput` ..... 64, 3021, 3022,  
     3024, 3025, 3110, 3197, 3202, 3259  
`\pdfprependkern` ..... 1338, 2905, 3363  
`\pdfprotrudechars` ..... 2871, 3206  
`\pdfstrcmp` ..... 417  
`\pdftexrevision` ..... 63, 68, 77, 79  
`\pdftexversion` 62, 66–68, 72, 74, 76, 84  
`\pickup@font` ..... 262,  
     1740, 1742, 1747, 1765, 1768, 1770  
`pifont` (package) ..... 66  
`protrusion` (option) ..... 5, 10, 94  
`PSNFSS` (package) ..... 148  
`pstricks` (package) ..... 7, 100  
`pxfonts` (package) ..... 16, 107
- Q**
- `qfonts` (package) ..... 16, 107, 156
- R**
- `reysize` (package) ..... 74  
`\rightmarginkern` ..... 199  
`\rprcode` ..... 753, 796, 797, 950, 954
- S**
- `selected` (option) .....  
     ..... 7, 13, 14, 22, 48, 60, 95, 154  
`\selectfont` ..... 1821, 1830, 3448  
`\set@fontsize` ..... 2041  
`\SetExpansion` ..... 7, 13, 2257, 3186  
`\SetExtraKerning` ..... 15, 2303, 3189  
`\SetExtraSpacing` ..... 14, 2282, 3188  
`\SetProtrusion` ..... 6, 11, 2236, 3185  
`\sfcode` ..... 3346  
`\shbscode` . 1172, 1173, 1186, 1191, 1199  
`\showhyphens` ..... 3314, 3318  
`shrink` (option) 6, 7, 14, 22, 97, 101, 156  
`slantsc` (package) ..... 129  
`soul` (package) ..... 149  
`spacing` (option) ..... 94, 95  
`statex` (package) ..... 30  
`\stbscode` . 1165, 1166, 1185, 1190, 1198  
`step` (option) .. 7, 22, 97, 101, 154, 156  
`stretch` (option) . 6, 7, 14, 18, 22, 97, 101
- T**
- `\textls` ..... 17, 265, 1353  
`tifa` (package) ..... 66  
`txfonts` (package) ..... 16, 107
- U**
- `unit` (option) ..... 6, 21, 155, 156  
`\UseMicrotypeSet` .....  
     .. 3, 10, 22, 35, 1893, 2112, 3184
- V**
- `verbose` (option) ..... 8, 22, 96, 155
- Z**
- `zefonts` (package) ..... 16, 107