



# 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)

v1.9d — 2006/05/05

## 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.

---

<sup>1</sup> A preview of the next version with support for even more micro-typographical extensions is also included in this package. Footnote 3 on page 4 contains the details.

<sup>2</sup> Currently, this package provides settings for Computer Modern Roman, Palatino, Times, Utopia Garamond, Adobe Garamond and Minion, Bitstream Charter, and the AMS math fonts, for various Euro symbol fonts, as well as some generic settings for unknown fonts. Contributions are very welcome.

## Contents

<b>1</b>	<b>Micro-Typography with pdfTeX</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>Selecting Fonts for Micro-Typography</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 . . . . .	14
5.4	Configuration Files . . . . .	15
<b>6</b>	<b>Context-sensitive Setup</b>	<b>16</b>
<b>7</b>	<b>Disabling Ligatures</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	Auxiliary macros . . . . .	29
13.3	Compatibility . . . . .	36
13.4	Setting up a font . . . . .	39
	[Protrusion – 43] [Expansion – 49] [Interword Space (Glue) – 51]	
	[Additional Kerning – 53] [Letterspacing – 55] [Disabling	
	Ligatures – 56] [Loading the Configuration – 57] [Translating Characters	
	into Slots – 61] [Hook into L <sup>A</sup> T <sub>E</sub> X's font selection – 66] [Context-sensitive	
	Setup – 68]	
13.5	Configuration . . . . .	70
	[Font Sets – 70] [Interaction with babel – 77] [Fine Tuning – 77] [Character	
	Inheritance – 83] [Permutation – 85]	

13.6 Package Options . . . . .	88
[Declaring the Options – 88] [Reading the Configuration File – 92] [Hook for Other Packages – 93] [Changing Options Later – 94] [Processing the Options – 96]	
<b>14 Configuration Files</b>	<b>103</b>
14.1 Font Sets . . . . .	103
14.2 Font Aliases . . . . .	103
14.3 Interaction with babel . . . . .	104
14.4 Note on Admissible Characters . . . . .	105
14.5 Character Inheritance . . . . .	105
[OT1 – 105] [T1 – 106] [LY1 – 106] [OT4 – 107] [QX – 107] [T5 – 108] [Euro symbols – 109]	
14.6 Font Expansion . . . . .	109
14.7 Character Protrusion . . . . .	110
[Default – 111] [Italics – 118] [Small Caps – 126] [Italic Small Caps – 129] [textcomp – 131] [Math – 136] [AMS fonts – 140] [Euro symbols – 145]	
14.8 Interword Spacing . . . . .	146
[Nonfrenchspacing – 148]	
14.9 Additional Kerning . . . . .	149
[French – 149] [Turkish – 150] [Letterspacing – 150]	
<b>15 Auxiliary File for Micro Fine Tuning</b>	<b>151</b>
<b>A Change History</b>	<b>153</b>
<b>B Index</b>	<b>160</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 pdfT<sub>E</sub>X

pdfT<sub>E</sub>X, the T<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:

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

Protrusion	off
Expansion	off

Both features are enabled in this document.

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

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

Both these features have been lacking a simple L<sup>A</sup>T<sub>E</sub>X user interface for quite some time. Then, the `pdfcprot` package was released, which allowed L<sup>A</sup>T<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 T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X users: Font expansion can now take place automatically. That is, pdfT<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.<sup>3</sup>

<sup>3</sup> pdfT<sub>E</sub>X version 1.40 (available from <http://sarovar.org/projects/pdfTEX/>) additionally implements two new experimental extensions: the *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. `microtype` already supports these new extensions, so that you can easily experiment with them. To generate the extended version of the `microtype` package and its documentation, simply remove the comments before `'\betatruer'` near the beginning of `microtype.ins` resp. `microtype.dtx`.

The microtype package provides an interface to all these micro-typographic extensions.<sup>4</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.

## 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}
```

<sup>4</sup> Therefore, it is an alternative, not a supplement, to the pdfcprot package, which provides an interface to character protrusion.

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

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 7.

## 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>5</sup>, you should either install the `cm-super` fonts or use the Latin Modern fonts (package `lmodern`).

<sup>5</sup> En passant, it may be noted that Type 1 format and T1 encoding are in no other way related than that both start with a 'T' and end with a '1'.

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

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

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

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

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

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

```
\usepackage{microtype}
```

**selected** true, false false

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

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

### 3.4 Miscellaneous Options

**DVIoutput** true, false false

pdfTeX is not only able to generate PDF output but can also spit out DVI files.<sup>6</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.

<sup>6</sup> TeX systems are beginning to switch to pdfTeX as the default engine even for DVI output.

- 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.
- 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 Selecting Fonts for Micro-Typography

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

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

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



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

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

If you now call

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

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

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

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

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

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

Sizes may be either specified as a dimension ('10' or '10pt'), or as a size selection command *without* the backslash. You may also specify ranges (e. g., 'small-Large'); while the lower boundary is included in the range, the upper boundary is not. Thus, '12-16' would match 12pt, 13.5pt, and 15.999pt, e. g., but not 16pt. You are allowed to omit the lower or upper bound ('-10', 'large-').

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

Table 1: Predefined font sets

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

‘\*’ = ‘default’

```
\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=basicmath]{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 ‘alltext’ font set will be used for character protrusion, the ‘basictext’ set for font expansion.

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 (and even UTF-8) characters may be entered directly or in L<sup>A</sup>T<sub>E</sub>X’s traditional 7-bit notation: both “A and Å are valid, provided the character is actually declared in both the input and the font encoding. You also have the possibility to declare lists of characters that should inherit protrusion or expansion factors (see section 5.3).

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 *font axis* = *value list* pairs as for the command `\DeclareMicrotypeSet` (see section 4). The only difference is that asterisked values will be expanded immediately instead of at the end of the preamble.

To find the matching settings for a given font the package will try all combinations of font encoding, family, series, shape and size, with decreasing significance in this order. For instance, if both settings for the current family (say, T1/cmr//) and settings for italic fonts in the normal weight (T1//m/it/) exist, those for the Computer Modern Roman font would apply.<sup>7</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>8</sup>

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

**context** The scope of the list may be limited to a certain context. For an example application, see section 6.

<sup>7</sup> For the interested, table 3 on page 59 presents the exact order.

<sup>8</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.

## 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  $\langle \text{character} \rangle = \langle \text{expansion factor} \rangle$  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><sub>X</sub> 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 typeset in 10pt:<sup>9</sup>

<sup>9</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.

```

\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 (even of different fonts) 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.

### 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 Å, Á, Â, Ã, Ä, Å and Æ should probably be protruded by the same (absolute) amount as the character A. Using the command `\DeclareCharacterInheritance`, you may declare such classes of characters, so that you then only have to set up the respective base character. With the optional argument, which may contain a comma-separated list

of features, you can confine the scope of the list. The font set can be declared in the usual way, with the only exception that exactly one encoding must be specified. The inheritance lists are 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, URW Garamond, Adobe Garamond and Minion<sup>10</sup>, for Bitstream Charter, the AMS math fonts and Euro symbols fonts (Adobe, ITC and Marvosym). 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 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, along with some others, in the default configuration file.

<sup>10</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, QX, (TS1) <sup>a</sup>	n, (it, sl, sc) <sup>a</sup>
Computer Modern Roman (cmr) <sup>b</sup>	OT1, OT4, T1, T5, LY1, TS1	n, it, sl, sc
Bitstream Charter (bch) <sup>c</sup>	OT1, T1, T5, LY1, TS1	n, it, (sl) <sup>d</sup> , sc
Adobe Garamond (pad, padx, padj)	OT1, T1, LY1, TS1	n, it, (sl) <sup>d</sup> , sc
URW Garamond (ugm) <sup>e</sup>	OT1, T1, TS1	n, it
Adobe Minion (pmnx, pmnj)	OT1, T1, LY1, TS1	n, it, (sl) <sup>d</sup> , sc, si
Palatino (ppl, pplx, pplj) <sup>f</sup>	OT1, OT4, T1, LY1, (TS1) <sup>a</sup>	n, it, (sl) <sup>d</sup> , sc
Times (ptm, ptmx, ptmj) <sup>g</sup>	OT1, OT4, T1, LY1, QX, (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
Euro fonts (Adobe, ITC, Marvosym)	U/OT1	n, it

<sup>a</sup> Incomplete

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

<sup>c</sup> Alias: mathdesign/Charter (mdbch)

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

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

<sup>f</sup> Aliases: pxfonts (pxr), qfonts/QuasiPalatino (qp1)

<sup>g</sup> Aliases: txfonts (txr), qfonts/QuasiTimes (qtm)

`\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>11</sup> This command will load the file `mt-<font name>.cfg`.

## 6 Context-sensitive Setup

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

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

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

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

<sup>11</sup> Font package authors might also want to have a look at the hook `\Microtype@Hook`, described in the implementation part, section 13.6.3.



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

and have microtype's context changed by the footnote marker command. This command differs among the various classes, here are some examples: for the base L<sup>A</sup>T<sub>E</sub>X classes, e.g., article:

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

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

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

For other classes, the command would have to be changed in a similar way.

## 7 Disabling Ligatures

```
\DisableLigatures {⟨set of fonts⟩}
```

A new feature has been introduced with pdfT<sub>E</sub>X 1.30: The possibility to completely disable all ligatures of a font (which will also switch off kerning for this font). While this purposely *lowers* the micro-typographic quality instead of raising it, it is especially useful for typewriter fonts, so that, e.g., in a T1 encoded font, ‘\texttt{--}’ will indeed be printed as ‘--’, not as ‘-’. \DisableLigatures may be used to specify, in the usual way, a set of fonts for which ligatures should be disabled, for example, of the typewriter font in T1 encoding:

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

## 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 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 L<sup>A</sup>T<sub>E</sub>X packages (except pdfcpot). However, life isn't perfect, so problems are to be expected. Currently, I am aware of the following issues:

- If you want to use 8-bit characters in the configuration, you have to load the `inputenc` package first. Unicode input is also supported (when loading `inputenc` with the `utf8` option, however, *not* with the `utf8x` option resp. the `ucs` package). If you are using multiple input encodings in your document, 8-bit characters will not work reliably.
- The CJK package, like `microtype`, hooks into the L<sup>A</sup>T<sub>E</sub>X font selection scheme. Therefore, both packages will currently not cooperate (`microtype` will simply overwrite CJK's definition). If you are interested in getting both packages to work together, and you can spare some time for testing, I would be glad to receive your feedback.

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

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

If you are using the `fancyvrb` or the `listings` package, this is not necessary, since their implementation of the corresponding environments will inhibit protrusion anyway.

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

- Warning: pdflatex: font ptmr8r cannot be expanded (not an included Type1 font)  
and the PDF viewer complains about a missing font, e. g., Adobe Reader:  
Could not find a font in the Resources dictionary - using Helvetica instead.  
Font expansion can only be applied if the font is actually embedded in the PDF file. If you receive the above error message, your  $\TeX$  system is not set up to embed (or ‘download’) the base PostScript fonts (e. g. Times, Helvetica, Courier). In most  $\TeX$  distributions, this can be changed in the file `updmap.cfg` by setting `pdfTeXDownloadBase14` to true. Otherwise, consult the local guide of your  $\TeX$  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 or TrueType 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 pdf $\TeX$  in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your  $\TeX$  system.
- ! 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, pdf $\TeX$  may be running out of memory. The memory can be increased by setting `pdf_mem_size` to a larger value (maximum 524 288). For te $\TeX$ -based systems, change the settings in `texmf.cnf`, for MiK $\TeX$ , in the file `miktex.ini`. Beginning with version 1.30 of pdf $\TeX$ , memory will grow dynamically, so that this problem can no longer occur.

## 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>12</sup>

<sup>12</sup> Should you have lots of pdfcprot configuration files lying around, I can also provide you with a  $\TeX$  conversion script. Just ask me.

## 10 Acknowledgments

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

*Harald Harders* has contributed protrusion settings for Adobe Minion. I would also like to thank him for a number of bug reports and suggestions he had to make. *Andreas Bühmann* has suggested the possibility to specify ranges of font sizes, and resourcefully assisted in implementing this. He also came up with some good ideas for the management of complex configurations. My thanks also go to *Maciej Eder* for contributing settings for the QX encoding.

I thank *Philipp Lehman* for adding to his csquotes package the possibility to restore the original meanings of all activated characters, thus allowing for these characters to be used in the configuration files. *Peter Wilson* kindly provided a hook in his ledmac/ledpar packages, so that critical editions can finally also benefit from character protrusion.

Additionally, the following people have reported bugs 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*, *Ralf Stubner*, *Holger Uhr* and *Peter Dyballa*.

## 11 References

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

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

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, *The pdfT<sub>E</sub>X user manual*, December 4, 2005. (Available from CTAN at [/systems/pdftex/manual/](#); latest version at <http://sarovar.org/projects/pdftex/>)

L<sup>A</sup>T<sub>E</sub>X3 Project Team, *L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub> font selection*, February 10, 2004. (Available from CTAN at [/macros/latex/doc/fntguide.pdf](#))

Carsten Schurig, Tobias Schlemmer, *The pdfcprot.sty package*, June 10, 2005. (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.

- 1.9d (5. 5. 2006)
  - Support for the Central European QX encoding (inheritance, generic protrusion settings, contributed by Maciej Eder; protrusion settings for Times)
  - Protrusion settings for various Euro symbol fonts (Adobe, ITC, Marvosym)
  - Support for Unicode input in the configuration (`inputenc/utf8`)
  - Compatibility with the `tex4ht` package
- 1.9c (2. 2. 2006)
  - Protrusion settings for URW Garamond
- 1.9b (20. 1. 2006)
  - Compatibility with the `listings` package
- 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 pdf $\TeX$  version 1.30 or later; see section 7)
  - New command `\microtypecontext` to change the configuration context; new key ‘context’ for the configuration commands (see section 6)
  - New key ‘font’ to add single fonts to the font sets (see section 4)
  - New key ‘preset’ to set all characters to the specified value before loading the lists
  - Value ‘relative’ renamed to ‘character’ for ‘unit’ keys
  - Support for the Polish OT4 encoding (protrusion, expansion, inheritance)
  - Support for the Vietnamese T5 encoding (protrusion, expansion, inheritance)
  - ‘DVOutput’ option will work with  $\TeX$ Live 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 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 pdf $\TeX$  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 13.6.3)

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 L $\TeX$  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.

beta: Support for features not yet included in an official release of pdf $\TeX$ .

config: Surrounds all configuration modules.

cfg-t: Surrounds latin text configurations.

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

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

cmr: Settings for Computer Modern Roman (mt-cmr.cfg).

pad: Settings for Adobe Garamond (mt-pad.cfg).

ppl: Settings for Palatino (mt-ppl.cfg).

ptm: Settings for Times (mt-ptm.cfg).

pmn: Settings for Adobe Minion (mt-pmn.cfg).

Contributed by Harald Harders.

ugm: Settings for URW Garamond (mt-ugm.cfg).

cfg-u: Surrounds non-text configurations (U encoding).

msa: Settings for AMS ‘a’ symbol font (mt-msa.cfg).

msb: Settings for AMS ‘b’ symbol font (mt-msb.cfg).

euf: Settings for AMS Euler Fraktur font (mt-euf.cfg).

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

cfg-e: Surrounds Euro symbol configurations.

zpeu: Settings for Adobe Euro symbol fonts (mt-zpeu.cfg).

euroitc: Settings for ITC Euro symbol fonts (mt-euroitc.cfg).

mvs: Settings for Marvosym Euro symbol (mt-mvs.cfg).

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

And now for something completely different.

1  $\langle *package \rangle$

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 pdf $\TeX$ .

```
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*\lststyle{}
18 \newcommand\textls[2] [] {#2}
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{%
27   \newcommand*#1{\MT@warning{%
28     \string#1 is deprecated. Please use\MessageBreak
29     \string#2 instead}%
30   \let #1#2#2}}

31 \MT@old@cmd\DeclareMicroTypeSet\DeclareMicrotypeSet
32 \MT@old@cmd\UseMicroTypeSet\UseMicrotypeSet
33 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
34 \MT@old@cmd\LoadMicroTypeFile\LoadMicrotypeFile

```

\MT@MT This is us.

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

\MT@error Communicate.

```

\MT@warning 36 \def\MT@error{\PackageError{\MT@MT}}
\MT@warning@n1 37 \def\MT@warning{\PackageWarning{\MT@MT}}
\MT@warn@err 38 \def\MT@warning@n1#1{\MT@warning{#1\@gobble}}
\MT@info 39 \def\MT@warn@err#1{\MT@error{#1}}{%
\MT@info 40   This error message appears because you loaded the ~\MT@MT'\MessageBreak
\MT@info@n1 41   package with the option ~verbose=errors'. Consult the documentation\MessageBreak
\MT@vinfo 42   in \MT@MT.(pdf,dvi) to find out what went wrong.}}
43 \def\MT@info{\PackageInfo{\MT@MT}}
44 \def\MT@info@n1#1{\MT@info{#1\@gobble}}
45 <debug>\let\MT@vinfo\@gobble

```

\tracingmicrotype Debug. Cases for \tracingmicrotype:

\MT@info 0: almost none

\MT@info@n1 1: + sets & lists

2: + heirs

3: + slots

4: + factors

```

46 <*debug>
47 \let\MT@vinfo\MT@info@n1
48 \newcount\tracingmicrotype
49 \tracingmicrotype=\tw@
50 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1\relax\else\MT@info{#2}\fi}
51 \def\MT@dinfo@n1#1#2{\ifnum\tracingmicrotype<#1\relax\else\MT@info@n1{#2}\fi}
52 </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>13</sup> ( $\geq$  1.40)

```

53 \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.

```

54 \ifx\normalpdftexversion\undefined \else
55   \let\pdftexversion\normalpdftexversion
56   \let\pdftexrevision\normalpdftexrevision
57   \let\pdfoutput\normalpdfoutput
58 \fi

```

Old packages might have let \pdftexversion to \relax.

```

59 \ifx\pdftexversion\undefined \else
60   \ifx\pdftexversion\relax \else
61 <debug> \MT@dinfo@n1{0}{running pdftex \the\pdftexversion(\pdftexrevision)}
62 <beta> \def\MT@pdftex@no{6}
63 <beta> \ifnum\pdftexversion < 140
64   \def\MT@pdftex@no{5}
65   \ifnum\pdftexversion < 130
66     \def\MT@pdftex@no{4}
67     \ifnum\pdftexversion < 120
68       \let\MT@pdftex@no\thr@
69       \ifnum\pdftexversion = 14

```

<sup>13</sup> This command was actually introduced in 1.30, however, pdfTeX ran in a buffer overflow with strings larger than 1024 bytes.

```

70      \ifnum \expandafter`\pdfTeXrevision < `h
71      \let\MT@pdfTeX@no\tw@
72      \ifnum \expandafter`\pdfTeXrevision < `f
73      \let\MT@pdfTeX@no@one
74      \fi
75      \fi
76      \else
77      \ifnum\pdfTeXversion < 14
78      \let\MT@pdfTeX@no@one
79      \fi
80      \fi
81      \fi
82      \fi
83 beta \fi
84 \fi
85 \fi
86 debug\MT@info@n1{0}{pdfTeX no: \number\MT@pdfTeX@no}

```

If we are not using pdf<sub>T</sub>E<sub>X</sub> or in case it is too old, we disable everything and exit here.

```

87 \ifnum\MT@pdfTeX@no<\tw@
88 \AtEndOfPackage{\let\@unprocessedoptions\relax}
89 \let\CurrentOption\@empty
90 \MT@warning@n1{%
91   \ifcase\MT@pdfTeX@no
92     You don't seem to be using pdfTeX.\MessageBreak
93   \or
94     You are using a pdfTeX version older than 0.14f.\MessageBreak
95     `MT@MT' won't work with such antiquated versions.\MessageBreak
96     Please install a newer version of pdfTeX.\MessageBreak
97   \fi
98   All micro-typographic features will be disabled}
99 \endinput\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.

```

100 \def\MT@catcodes{%
101   \catcode`\^7 %
102   \@makeother\~%
103   \@makeother\=%
104   \@makeother\*%
105   \@makeother\,%
106   \@makeother\/%
107   \@makeother\`%
108   \@makeother\'%
109   \@makeother\"%
110   \@makeother\!%
111 }

```

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

```

112 \def\MT@restore@catcodes#1{%
113   \ifx\relax#1\else
114     \noexpand\catcode`\noexpand#1\the\catcode`#1\relax
115     \expandafter\MT@restore@catcodes
116   \fi
117 }
118 \edef\MT@restore@catcodes{\MT@restore@catcodes\^-\=\*\,\/\\`\'\"!\relax}

```

```

119 \MT@catcodes
120 \AtEndOfPackage{\MT@restore@catcodes}

```

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

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

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

```
122 \newtoks\MT@toks
```

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

```
\ifMT@expansion 123 \newif\ifMT@protrusion
```

```
\ifMT@auto 124 \newif\ifMT@expansion
```

```
125 \newif\ifMT@auto
```

```
\ifMT@selected 126 \newif\ifMT@selected
```

```
\ifMT@noligatures 127 \newif\ifMT@noligatures
```

```
\ifMT@draft 128 \newif\ifMT@draft
```

```
129 \beta
```

```
\ifMT@spacing 130 \newif\ifMT@spacing
```

```
\ifMT@kerning 131 \newif\ifMT@kerning
```

```
\ifMT@babel 132 \newif\ifMT@babel
```

```
133 /\beta
```

`\ifMT@document` Our private test whether we’re still in the preamble.

```
134 \newif\ifMT@document
```

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

```
\MT@pr@factor 135 \let\MT@pr@level\tw@
```

```
\MT@pr@unit 136 \let\MT@pr@factor\@m
```

```
137 \let\MT@pr@unit\@empty
```

```
\MT@ex@level 138 \let\MT@ex@level\tw@
```

```
\MT@ex@factor 139 \let\MT@ex@factor\@m
```

```
\MT@stretch 140 \let\MT@stretch\m@ne
```

```
141 \let\MT@shrink \m@ne
```

```
\MT@shrink 142 \let\MT@step \m@ne
```

```
\MT@step 143 \beta
```

```
144 \let\MT@sp@factor\@m
```

```
145 \let\MT@kn@factor\@m
```

`\MT@sp@factor` Default unit for spacing settings is space, default unit for kerning is 1em.

```
\MT@kn@factor 146 \let\MT@sp@unit\m@ne
```

```
\MT@kn@unit 147 \def\MT@kn@unit{1em}
```

```
\MT@letterspacing 148 \let\MT@letterspacing\m@ne
```

```
149 /\beta
```

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

```
\MT@pr@max 150 \def\MT@pr@min{-\@m}
```

```
\MT@ex@min 151 \let\MT@pr@max\@m
```

```
152 \let\MT@ex@min\z@
```

```
\MT@ex@max 153 \let\MT@ex@max\@m
```

```
\MT@sp@min 154 \beta
```

```
155 \def\MT@sp@min{-\@m}
```

```
\MT@sp@max 156 \let\MT@sp@max\@m
```

```
\MT@kn@min 157 \def\MT@kn@min{-\@m}
```

```
158 \let\MT@kn@max\@m
```

```
159 /\beta
```

`\MT@factor@default` Default factor.

```
160 \def\MT@factor@default{1000 }
```

`\MT@stretch@default` Default values for expansion.

```
\MT@shrink@default
```

```
\MT@step@default
```

```

161 \def\MT@stretch@default{20 }
162 \def\MT@shrink@default{20 }
163 \def\MT@step@default{4 }

```

`\MT@letterspacing@default` Default value for letterspacing (in thousandths of 1em).

```

164 <beta> \def\MT@letterspacing@default{100 }

```

## 13.2 Auxiliary macros

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

```

165 \expandafter\let\expandafter\MT@requires@etex
166   \ifcase 0%
167     \ifx\TeXversion\undefined 1\else
168       \ifx\TeXversion\relax 1\else
169         \ifcase\TeXversion 1\fi
170       \fi
171     \fi\space
172     \@firstoftwo
173   \else
174     \@secondoftwo
175   \fi
176 <debug> \MT@info{n}{0}{\MT@requires@etex}{\not }running etex

```

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

```

177 \def\MT@requires@pdftex#1{%
178   \ifnum\MT@pdftex@no<#1\relax
179     \expandafter\@secondoftwo
180   \else
181     \expandafter\@firstoftwo
182   \fi
183 }

```

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

```

184 \def\MT@def@n#1{\expandafter\def\csname #1\endcsname}

```

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

```

185 \def\MT@edef@n#1{\expandafter\edef\csname #1\endcsname}

```

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

```

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

```

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

```

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

```

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

```

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

```

`\MT@exp@string` Remove trailing space.

```

189 \def\MT@exp@string{\expandafter\string}

```

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

```

190 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}

```

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

```

191 \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.

```

192 \def\MT@exp@two@n#1#2#3{%

```

```

193 \expandafter\expandafter\expandafter
194   #1\expandafter\expandafter\expandafter
195   {\expandafter#2\expandafter}\expandafter{#3}}

```

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

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

```

\MT@ifdefined@c@T 196 \MT@requires@etex{
197   \def\MT@ifdefined@c@T#1{%
198     \ifdefined#1%
199     \expandafter\@firstofone
200   \else
201     \expandafter\@gobble
202   \fi
203 }
204 \def\MT@ifdefined@c@TF#1{%
205   \ifdefined#1%
206   \expandafter\@firstoftwo
207 \else
208   \expandafter\@secondoftwo
209 \fi
210 }
211 \def\MT@ifdefined@n@T#1{%
212   \ifcsname#1\endcsname
213   \expandafter\@firstofone
214 \else
215   \expandafter\@gobble
216 \fi
217 }
218 \def\MT@ifdefined@n@TF#1{%
219   \ifcsname#1\endcsname
220   \expandafter\@firstoftwo
221 \else
222   \expandafter\@secondoftwo
223 \fi
224 }
225 }{
226 \def\MT@ifdefined@c@T#1{%
227   \ifx#1\@undefined
228   \expandafter\@gobble
229 \else
230   \expandafter\@firstofone
231 \fi
232 }
233 \def\MT@ifdefined@c@TF#1{%
234   \ifx#1\@undefined
235   \expandafter\@secondoftwo
236 \else
237   \expandafter\@firstoftwo
238 \fi
239 }
240 \def\MT@ifdefined@n@T#1{%
241   \begingroup\MT@exp@two@c\endgroup
242   \ifx\csname #1\endcsname\relax
243   \expandafter\@gobble
244 \else
245   \expandafter\@firstofone
246 \fi
247 }

```

```

248 \def\MT@ifdefined@n@TF#1{%
249 \begingroup\MT@exp@two@c\endgroup
250 \ifx\csname #1\endcsname\relax
251 \expandafter\@secondoftwo
252 \else
253 \expandafter\@firstoftwo
254 \fi
255 }
256 }

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

257 \MT@requires@etex{
258 \def\MT@detokenize@n#1{\detokenize\expandafter{\string#1}}
259 \def\MT@detokenize@c#1{\detokenize
260 \expandafter\expandafter\expandafter{\expandafter\string#1}}
261 }{
262 \def\MT@detokenize@n#1{\string#1}
263 \def\MT@detokenize@c#1{\MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty}
264 }

\MT@ifempty Test whether argument is empty.

265 \begingroup
266 \catcode`\%=12
267 \catcode`\&=14
268 \gdef\MT@ifempty#1{%
269 \if %#1%&
270 \expandafter\@firstoftwo
271 \else
272 \expandafter\@secondoftwo
273 \fi
274 }
275 \endgroup

\MT@ifint Test whether argument is an integer, using an old trick by Mr. Arseneau, or the
latest and greatest from pdfTeX.

276 \MT@requires@pdfTeX{
277 \def\MT@ifint#1{%
278 \ifcase\pdfmatch{^[0-9]+ *$}{#1}\relax
279 \expandafter\@secondoftwo
280 \else
281 \expandafter\@firstoftwo
282 \fi
283 }
284 }{
285 \def\MT@ifint#1{%
286 \if!\ifnum9<1#1!\else?\fi
287 \expandafter\@firstoftwo
288 \else
289 \expandafter\@secondoftwo
290 \fi
291 }
292 }

\MT@ifdimen Test whether argument is dimension (or number).

293 \MT@requires@pdfTeX{
294 \def\MT@ifdimen#1{%
295 \ifcase\pdfmatch{^([0-9]+([.,][0-9]+)?|[.,][0-9]+)%
296 (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc)? *$}{#1}\relax

```

```

297     \expandafter\@secondoftwo
298     \else
299     \expandafter\@firstoftwo
300     \fi
301   }
302 }{
303   \def\MT@ifdimen#1{%
304     \setbox\z@=\hbox{%
305       \MT@count=1#1\relax
306       \ifnum\MT@count=\@ne
307         \aftergroup\@secondoftwo
308       \else
309         \aftergroup\@firstoftwo
310       \fi
311     }%
312   }
313 }

```

**\MT@ifdim** Test floating point numbers.

```

314 \def\MT@ifdim#1#2#3{%
315   \ifdim #1\p@ #2 #3\p@
316     \expandafter\@firstoftwo
317   \else
318     \expandafter\@secondoftwo
319   \fi
320 }

```

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

```

321 \MT@requires@pdftex5{
322   \def\MT@ifstreq#1#2{%
323     \ifcase\pdfstrcmp{#1}{#2}\relax
324     \expandafter\@firstoftwo
325   \else
326     \expandafter\@secondoftwo
327   \fi
328   }
329 }{
330   \def\MT@ifstreq#1#2{%
331     \edef\x{#1}%
332     \edef\y{#2}%
333     \ifx\x\y
334       \expandafter\@firstoftwo
335     \else
336       \expandafter\@secondoftwo
337     \fi
338   }
339 }

```

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

```

340 \def\MT@xadd#1#2{%
341   \ifx#1\relax
342     \xdef#1{#2}%
343   \else
344     \xdef#1{#1#2}%
345   \fi
346 }

```

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

```

347 \def\MT@xaddb#1#2{%
348   \ifx#1\relax
349     \xdef#1{#2}%

```



```

350 \else
351 \xdef#1{#2#1}%
352 \fi
353 }

\MT@map@clist@n Run <#2> on all elements of the comma list <#1>. This and the following is modelled
\MT@map@clist@c after LATEX3 commands.
\MT@map@clist@ 354 \def\MT@map@clist@n#1#2{%
\MT@clist@function 355 \ifx\@empty#1\else
\MT@clist@break 356 \def\MT@clist@function##1{#2}%
357 \MT@map@clist@#1,\@nil,\@nnil
358 \fi
359 }
360 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
361 \def\MT@map@clist@#1,{%
362 \ifx\@nil#1%
363 \expandafter\MT@clist@break
364 \fi
365 \MT@clist@function{#1}%
366 \MT@map@clist@
367 }
368 \let\MT@clist@function\@gobble
369 \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
\MT@map@tlist@c to jump out of the loop.
\MT@map@tlist@ 370 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break 371 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
372 \def\MT@map@tlist@#1#2{%
373 \ifx\@nnil#2\else
374 #1{#2}%
375 \expandafter\MT@map@tlist@
376 \expandafter#1%
377 \fi
378 }
379 \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 380 \newif\ifMT@inlist@
381 \def\MT@in@clist#1#2{%
382 \def\x##1,#1,##2##3\@nnil{%
383 \ifx##2\@empty
384 \MT@inlist@false
385 \else
386 \MT@inlist@true
387 \fi
388 }%
389 \expandafter\x\expandafter,#2,#1,\@empty\@nnil
390 }

\MT@rem@from@clist Remove item <#1> from comma list <#2>. Doesn't work if the item to be removed
is included more than once in the list. Using \pdfmatch and \pdflastmatch here
would be really slow!
391 \def\MT@rem@from@clist#1#2{%
392 \def\x##1,#1,##2##3\@nnil{%
393 \ifx##2\@empty\else
394 \ifx\\##1\\global\let#2\@gobble
395 \else\xdef#2{\@gobble##1}\fi
396 \def\x###1,#1,###2###3\@nnil{\xdef#2{#2###1}}%
397 \x,##2##3,#1,\@nnil

```

```

398   \fi
399 }%
400 \expandafter\x\expandafter,#2,#1,\@empty\@nnil
401 }

\MT@in@tlist Test whether item is in token list. Since this isn't too elegant, I thought that at least
\MT@in@tlist@ here, \pdfmatch would be more efficient – however, it turned out to be even slower
than this solution.
402 \def\MT@in@tlist#1#2{%
403   \MT@inlist@false
404   \def\x{#1}%
405   \MT@map@tlist@c#2\MT@in@tlist@
406 }
407 \def\MT@in@tlist@#1{%
408   \edef\y{#1}%
409   \ifx\x\y
410     \MT@inlist@true
411     \expandafter\MT@tlist@break
412   \fi
413 }

\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@@ 414 \def\MT@in@rlist#1{%
\MT@size@name 415   \MT@inlist@false
416   \MT@map@tlist@c#1\MT@in@rlist@
417 }
418 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
419 \def\MT@in@rlist@@#1#2#3{%
420   \MT@ifdim{#2}=\m@ne{%
421     \MT@ifdim{#1}=\MT@size
422     \MT@inlist@true
423     \relax
424   }{%
425     \MT@ifdim\MT@size<{#1}\relax{%
426       \MT@ifdim\MT@size<{#2}%
427       \MT@inlist@true
428       \relax
429     }%
430   }%
431   \ifMT@inlist@
432     \def\MT@size@name{#3}%
433     \expandafter\MT@tlist@break
434   \fi
435 }

\MT@loop This is the same as LATEX's \loop, which we mustn't use, since this could confuse an
\MT@iterate outer \loop in the document.
\MT@repeat 436 \def\MT@loop#1\MT@repeat{%
437   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
438   \MT@iterate \let\MT@iterate\relax
439 }
440 \let\MT@repeat\fi

\MT@while@num Execute <#3> from <#1> up to (excluding) <#2>.
441 \def\MT@while@num#1#2#3{%
442   \@tempcnta#1\relax
443   \MT@loop #3%
444   \advance\@tempcnta \@ne
445   \ifnum\@tempcnta < #2\MT@repeat

```

```

446 }

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

\MT@count Increment macro <#1> by one. Saves using up too many counters. The e-TeX way is
\MT@increment slightly faster.
448 \newcount\MT@count
449 \MT@requires@etex{
450   \def\MT@increment#1{\edef#1{\number\numexpr #1 + 1\relax}}
451 }{
452   \def\MT@increment#1{%
453     \MT@count=#1\relax
454     \advance\MT@count \@ne
455     \edef#1{\number\MT@count}%
456   }
457 }

\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.
458 \MT@requires@etex{
459   \def\MT@scale#1#2#3{%
460     \ifnum #3 = \z@
461       #1=\numexpr #1 * #2\relax
462     \else
463       #1=\numexpr #1 * #2 / #3\relax
464     \fi
465   }
466 }{
467   \def\MT@scale#1#2#3{%
468     \multiply #1 #2\relax
469     \ifnum #3 = \z@ \else
470       \divide #1 #3\relax
471     \fi
472   }
473 }

\MT@make@string Set the category code of all characters to 12.
474 \let\MT@make@string\@onelevel@sanitize

\MT@abbr@pr Some abbreviations. Thus, we can have short command names but full-length log
\MT@abbr@ex output.
\MT@abbr@pr@c 475 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 476 \def\MT@abbr@ex{expansion}
\MT@abbr@pr@inh 477 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@ex@inh 478 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh 479 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@nl 480 \def\MT@abbr@ex@inh{expansion inheritance}
481 \def\MT@abbr@nl{noligatures}

\MT@abbr@sp Also for the experimental features.
\MT@abbr@sp@c 482 <*beta>
\MT@abbr@sp@inh 483 \def\MT@abbr@sp{spacing}
\MT@abbr@kn@c 484 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@kn@inh 485 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@kn@c 486 \def\MT@abbr@kn{kerning}
\MT@abbr@kn@inh 487 \def\MT@abbr@kn@c{kerning codes}

```

```

488 \def\MT@abbr@kn@inh{kerning inheritance}
489 </beta>

\MT@rbb@protrusion These we also need the other way round.
\MT@rbb@expansion 490 \def\MT@rbb@protrusion{pr}
\MT@rbb@spacing    491 \def\MT@rbb@expansion{ex}
\MT@rbb@kerning    492 < *beta>
                   493 \def\MT@rbb@spacing{sp}
                   494 \def\MT@rbb@kerning{kn}
                   495 </beta>

\MT@features We can work on these lists to save some guards in the dtx file.
\MT@features@long 496 \def\MT@features{pr,ex%
                   497 <beta>,sp,kn%
                   498 }
                   499 \def\MT@features@long{protrusion,expansion%
                   500 <beta>,spacing,kerning%
                   501 }

\MT@is@feature Whenever an optional argument accepts a list of features, we can use this com-
                 mand to check whether a feature exists in order to prevent a rather confusing
                 ‘Missing \endcsname inserted’ error message. The feature (long form) must be in
                 \@tempa, the type of list to ignore in <#1>, then comes the action.
                 502 \def\MT@is@feature#1{%
                 503   \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
                 504   \ifMT@inlist@
                 505     \expandafter\@firstofone
                 506   \else
                 507     \MT@error{\@tempa' is not an available micro-typographic\MessageBreak
                 508       feature. Ignoring #1}{Available features are: \MT@features@long'.}%
                 509     \expandafter\@gobble
                 510   \fi
                 511 }

```

### 13.3 Compatibility

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

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

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

```
512 \let\MT@setup@\@empty
```

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

```

513 \def\MT@addto@setup#1{%
514   \begingroup
515     \MT@toks\expandafter{\MT@setup@#1}%
516     \xdef\MT@setup@{\the\MT@toks}%
517   \endgroup
518 }

```

It will be executed at the end of the preamble.

```

519 \AtBeginDocument{\MT@setup@}

\MT@with@package We almost never do anything if a package is not loaded.
520 \def\MT@with@package#1{\ifpackageloaded{#1}\@firstofone\@gobble}

\MT@pdfcp@error Our competitor, the pdfcp@ package, must not be tolerated!
521 \def\MT@pdfcp@error{%
522   \MT@error{Detected the `pdfcp@' package!\MessageBreak
523     `~\MT@MT' and `pdfcp@' may not be used together}{%
524   The `pdfcp@' package provides an interface to character protrusion.\MessageBreak
525   So does the `~\MT@MT' package. Using both packages at the same\MessageBreak
526   time will almost certainly lead to undesired results. Have your choice!}%
527   \let\MT@pdfcp@error\relax
528 }
529 \MT@with@package{pdfcp@}\MT@pdfcp@error

\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).

530 \def\MT@ledmac@setup{%
531   \ifMT@protrusion
532     \MT@requires@pdftex5{%
533       \MT@ifdefined@c@TF\l@dunhbox@line{%
534         \MT@info@nl{Patching ledmac to enable character protrusion}%
535         \newdimen\MT@led@kern
536         \let\MT@led@unhbox@line\l@dunhbox@line
537         \renewcommand*{\l@dunhbox@line}[1]{%
538           \ifhbox##1%
539             \MT@led@kern=\rightmarginkern##1%
540             \kern\leftmarginkern##1%
541             \MT@led@unhbox@line##1%
542             \kern\MT@led@kern
543           \fi
544         }%
545       }%
546       \MT@warning@nl{%
547         Character protrusion in paragraphs with line\MessageBreak
548         numbering will only work if you update ledmac}%
549     }%
550   }%
551   \MT@warning@nl{%
552     The pdftex version you are using does not allow\MessageBreak
553     character protrusion in paragraphs with line\MessageBreak
554     numbering by the `ledmac' package.\MessageBreak
555     Upgrade pdftex to version 1.30 or later}%
556   }%
557   \fi
558 }

\MT@restore@p@h Restore meaning of \# and \%.
559 \def\MT@restore@p@h{\chardef\%`% \chardef\#`# }
```

`\MT@setupfont@hook` This hook will be executed every time a font is set up (inside a group).

In the preamble, we check for the packages each time a font is set up. Thus, it will work regardless when the packages are loaded.

Even for packages that don't activate any characters in the preamble (like `babel` and `csquotes`), we have to check here, too, in case they were loaded before `microtype`, and a font is loaded `\AtBeginDocument`, before `microtype`. (This is no longer needed, since the complete setup is now deferred until the end of the preamble. However, it is still necessary for `defersetup=false`.)

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

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

```
561 \ifpackagewith{babel}{spanish}{%
562   \MT@ifdefined@c@T\percentsign{\let%\percentsign}%
563 } \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.)

```
564 \MT@with@package{csquotes}{%
565   \@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax
566 }%
```

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

```
567 \ifpackageloaded{hyperref}\MT@restore@p@h{%
568   \MT@with@package{tex4ht}\MT@restore@p@h
569 }%
570 }
```

Check again at the end of the preamble.

```
571 \MT@addto@setup{%
572   \MT@with@package{pdfcpot}\MT@pdfcpot@error
573   \MT@with@package{ledmac}\MT@ledmac@setup
```

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

```
574 \let\MT@setupfont@hook\empty
575 \ifpackagewith{babel}{spanish}{%
576   \g@addto@macro\MT@setupfont@hook{%
577     \MT@ifdefined@c@T\percentsign{\let%\percentsign}}%
578 } \relax
579 \MT@with@package{csquotes}{%
580   \@ifpackagelater{csquotes}{2005/05/11}{%
581     \g@addto@macro\MT@setupfont@hook\@disablequotes
582   }{%
583     \MT@warning@n1{%
584       Should you receive warnings about unknown slot\MessageBreak
585       numbers, try upgrading the `csquotes' package}%
586   }%
587 }%
588 \ifpackageloaded{hyperref}{%
589   \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
```

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

```
590 \pdfstringdefDisableCommands{%
591   \let\pickup@font\MT@orig@pickupfont
592 } *beta
```

```

593     \let\lsstyle\empty
594     \def\textls#1#\@firstofone{%
595 \beta
596     }%
597 }{%
598     \MT@with@package{tex4ht}{%
599     \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
600     }%
601 }%

```

The listings package makes numbers and letters active.

```

602 \MT@with@package{listings}{%
603     \g@addto@macro\MT@begin@catcodes{%
604     \MT@while@num{"30"}{"3A"}{\catcode\@tempcnta 12\relax}%
605     \MT@while@num{"41"}{"5B"}{\catcode\@tempcnta 11\relax}%
606     \MT@while@num{"61"}{"7B"}{\catcode\@tempcnta 11\relax}%
607     }%

```

When loaded with the `extendedchar` option, listings will also redefine 8-bit active characters (`inputenc`). Luckily, this simple redefinition will make them expand to their original definition, so that they could be used in the configuration.

```

608     \g@addto@macro\MT@setupfont@hook{%
609     \let\lst@ProcessLetter\empty
610     }%
611 }%

```

Of course, using both soul's and microtype's letterspacing mechanisms at the same time doesn't make much sense. But soul can do more, e. g., underlining. The optional argument to `\textls` may not be used.

```

612 \beta
613 \MT@with@package{soul}{%
614     \soulregister\lsstyle 0%
615     \soulregister\textls 1%
616     }%
617 \beta
618 }

```

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

```

619 \expandafter\ifx\the\font\nullfont\normalfont\fi

```

## 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.

```

620 \def\MT@setupfont{%
621     \ifx\MT@vinfo\MT@info\nl
622     \MT@info{Setting up font ~\MT@exp@string\MT@font'}\fi

```

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

```

623     \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!

```

624     \MT@exp@two@c\MT@split@name\string\MT@font\@ni1

```

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

```

625 \MT@exp@one@n\MT@find@file\MT@family
626 \ifx\MT@familyalias\empty \else
627 \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.)

```

628 \ifx\fontencoding\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!

```

629 \MT@protrusion
630 \MT@expansion

```

Interword spacing and kerning.

```

631 \beta
632 \MT@spacing
633 \MT@kerning
634 /beta

```

Disable ligatures?

```

635 \MT@noligatures
636 }

```

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

```

\MT@encoding 637 \def\MT@split@name#1/#2/#3/#4/#5\@nil{%
\MT@family   638 \def\MT@encoding{#1}%
\MT@family   639 \def\MT@family{#2}%
\MT@series   640 \def\MT@series{#3}%
\MT@shape    641 \def\MT@shape{#4}%
\MT@size     642 \def\MT@size{#5}%

```

`\MT@familyalias` Alias family?

```

643 \MT@ifdefined@n@TF{MT@MT@family @alias}%
644 {\MT@let@cn\MT@familyalias{MT@MT@family @alias}}%
645 {\let\MT@familyalias\empty}%
646 }

```

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

```

\MT@feat
\MT@maybe@do 647 \newif\ifMT@do
648 \def\MT@maybe@do#1{%

```

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

```

649 \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).

```

650 \MT@dotrue
651 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
652 \MT@ifdefined@n@TF{MT@checklist@##1}%
653 {\csname MT@checklist@##1\endcsname}%
654 {\MT@checklist@{##1}}%
655 {#1}%
656 }%

```



```

657 \else
658 \MT@dofalse
659 \fi
660 \ifMT@do

```

\MT@feat stores the current feature.

```

661 \def\MT@feat{#1}%
662 \csname MT@set@#1@codes\endcsname
663 \else
664 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
665 \fi
666 }

```

\MT@checklist@ The generic test.

```

667 \def\MT@checklist@#1#2{%
668 \edef\@tempa{\csname MT@#2@setname\endcsname}%
669 \!debug \MT@ifdefined@n@T
670 \!debug \MT@ifdefined@n@TF
671 {MT@#2list@#1@\@tempa}{%

```

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

```

672 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
673 \csname MT@#1\expandafter\endcsname
674 \csname MT@#2list@#1@\@tempa\endcsname
675 \ifMT@inlist@
676 \!debug \MT@dinfo@n{1}{\@nameuse{MT@abbr@#2}: #1 ~\@nameuse{MT@#1}' in list}%
677 \MT@dotrue
678 \else
679 \!debug \MT@dinfo@n{1}{\@nameuse{MT@abbr@#2}: #1 ~\@nameuse{MT@#1}' not in list}%
680 \MT@dofalse
681 \expandafter\MT@clist@break
682 \fi
683 }%

```

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

```

684 \!debug {\MT@dinfo@n{1}{\@nameuse{MT@abbr@#2}: #1 list empty}}%
685 }

```

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

```

686 \def\MT@checklist@font#1{%
687 \edef\@tempa{\csname MT@#1@setname\endcsname}%
688 \!debug \MT@ifdefined@n@T
689 \!debug \MT@ifdefined@n@TF
690 {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.

```

691 \expandafter\expandafter\expandafter\MT@exp@one@n
692 \expandafter\expandafter\expandafter\MT@in@clist
693 \expandafter\expandafter\expandafter{%
694 \expandafter\expandafter\expandafter\string
695 \expandafter\MT@font\expandafter}%
696 \csname MT@#1list@font@\@tempa\endcsname
697 \ifMT@inlist@
698 \!debug \MT@dinfo@n{1}{\@nameuse{MT@abbr@#1}: font ~\MT@font' in list}%
699 \expandafter\MT@clist@break
700 \else
701 \!debug \MT@dinfo@n{1}{\@nameuse{MT@abbr@#1}: font ~\MT@font' not in list}%

```

```

702     \MT@dofalse
703     \fi
704 }%
705 <debug> {\MT@info@n1}{\@nameuse{MT@abbr@#1}: font list empty}}%
706 }

```

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

```

707 \def\MT@checklist@family#1{%
708   \edef\@tempa{\csname MT@#1@setname\endcsname}%
709   <!debug> \MT@ifdefined@nT
710   <debug> \MT@ifdefined@nTF
711   {MT@#1list@family@\@tempa}{%
712     \MT@exp@two@n\MT@in@clist
713     \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
714     \ifMT@inlist@
715     <debug> \MT@info@n1{\@nameuse{MT@abbr@#1}: family `\'@nameuse{MT@family}' in list}%
716     \MT@dotrue
717   }else
718   <debug> \MT@info@n1{\@nameuse{MT@abbr@#1}: family `\'@nameuse{MT@family}' not in list}%
719   \MT@dofalse
720   \ifx\MT@familyalias\@empty \else
721     \MT@exp@two@n\MT@in@clist
722     \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
723     \ifMT@inlist@
724     <debug> \MT@info@n1{\@nameuse{MT@abbr@#1}: alias `\'@nameuse{MT@familyalias}' in list}%
725     \MT@dotrue
726   <debug> \else \MT@info@n1{\@nameuse{MT@abbr@#1}: alias `\'@nameuse{MT@familyalias}' not in list}%
727   \fi
728   \fi
729   \fi
730   \ifMT@do \else
731     \expandafter\MT@clist@break
732   \fi
733 }%
734 <debug> {\MT@info@n1}{\@nameuse{MT@abbr@#1}: family list empty}}%
735 }

```

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

```

736 \def\MT@checklist@size#1{%
737   \edef\@tempa{\csname MT@#1@setname\endcsname}%
738   <!debug> \MT@ifdefined@nT
739   <debug> \MT@ifdefined@nTF
740   {MT@#1list@size@\@tempa}{%
741     \expandafter\MT@in@rlist
742     \csname MT@#1list@size@\@tempa\endcsname
743     \ifMT@inlist@
744     <debug> \MT@info@n1{\@nameuse{MT@abbr@#1}: size `\'@nameuse{MT@size}' in list}%
745     \MT@dotrue
746   }else
747   <debug> \MT@info@n1{\@nameuse{MT@abbr@#1}: size `\'@nameuse{MT@size}' not in list}%
748   \MT@dofalse
749   \expandafter\MT@clist@break
750   \fi
751 }%
752 <debug> {\MT@info@n1}{\@nameuse{MT@abbr@#1}: size list empty}}%
753 }

```

### 13.4.1 Protrusion

`\MT@protrusion` Set up for protrusion?

```
754 \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.

```
755 \def\MT@set@pr@codes{%
    Check whether and if, which list should be applied to the current font.
    756 \MT@if@list@exists{%
    757 \MT@get@dimen@six
    758 \MT@get@opt
    759 \MT@reset@pr@codes

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

    Load additional lists?
    761 \MT@load@list\MT@pr@c@name
    762 \MT@set@listname

    Load the main list.
    763 \MT@let@cn\@tempc{\MT@pr@c@\MT@pr@c@name}%
    764 \expandafter\MT@pr@do\@tempc,\relax,%
    765 }\MT@reset@pr@codes
    766 }

    \MT@set@all@pr Set all protrusion codes of the font.
    767 \def\MT@set@all@pr#1#2{%
    768 <debug>\MT@info@nl{3}{-- lp/rp: setting all to \number#1/\number#2}%
    769 \MT@do@font{%
    770 \lcode\MT@font\@tempcnta=#1\relax
    771 \rcode\MT@font\@tempcnta=#2\relax
    772 }%
    773 }

    \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.
    774 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
    775 \let\MT@reset@pr@codes\relax

    \MT@gobble@settings If \fontdimen6 is zero, character protrusion won't work, and we can skip the
    \MT@dimen@six settings (for example, the dsfont fonts don't specify this dimension; this is probably
    \MT@get@dimen@six a bug).
    776 \def\MT@get@dimen@six{%
    777 \ifnum\fontdimen6\MT@font=\z@
    778 \MT@warning@nl{%
    779 Font '\MT@exp@string\MT@font' does not specify its\MessageBreak
    780 \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
    781 \@nameuse{\MT@abbr@\MT@feat} will not work with this font}%
    782 \expandafter\MT@gobble@settings
    783 \else
    784 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
    785 \fi
    786 }
    787 \def\MT@gobble@settings#1\@tempc,\relax,{}%
```

`\MT@pr@do` Split up the values and set `\lpcode` and `\rpcodes`.

```
788 \def\MT@pr@do#1,{%
789   \ifx\relax#1\@empty\else
790     \MT@pr@split #1=\relax
791     \expandafter\MT@pr@do
792   \fi
793 }
```

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

```
794 \def\MT@pr@split#1=#2=#3\relax{%
795   \def\@tempa{#1}%
796   \ifx\@tempa\@empty \else
797     \MT@get@slot
798     \ifnum\MT@char > \m@ne
799       \MT@get@char@unit
800       \MT@pr@split@val#2\relax
801     \fi
802   \fi
803 }
```

`\MT@pr@split@val`

```
804 \def\MT@pr@split@val#1,#2\relax{%
805   \def\@tempb{#1}%
806   \MT@ifempty\@tempb\relax{%
807     \MT@scale@to@em
808     \lpcode\MT@font\MT@char=\@tempcntb
809     (debug) \MT@dinfo@n1{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char: [#1]}%
810   }%
811   \def\@tempb{#2}%
812   \MT@ifempty\@tempb\relax{%
813     \MT@scale@to@em
814     \rpcodes\MT@font\MT@char=\@tempcntb
815     (debug) \MT@dinfo@n1{4}{;;; rp (\MT@char): \number\rpcodes\MT@font\MT@char: [#2]}%
816   }%
```

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

```
817 \MT@ifdefined@c@T\MT@pr@inh@name{%
818   \MT@ifdefined@n@T\MT@inh@\MT@pr@inh@name @\MT@char @}{%
819     \expandafter\MT@map@tlist@c
820     \csname MT@inh@\MT@pr@inh@name @\MT@char @\endcsname
821     \MT@set@pr@heirs
822   }%
823 }%
824 }
```

`\MT@scale@to@em` Since pdf<sub>T</sub>E<sub>X</sub> 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 `\lpcode` resp. `\rpcodes`, since this would disallow protrusion factors larger than the character width (since `\[l]rcode`’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`.

```

825 \MT@requires@pdfTeX{
826   \def\MT@scale@to@em{%
827     \@tempcntb=\MT@count\relax

```

For really huge fonts (100pt or so), an arithmetic overflow could occur with vanilla T<sub>E</sub>X. Using e-T<sub>E</sub>X, this can't happen, since the intermediate value is 64 bit, which could only be reached with a character width larger than \maxdimen.

```

828   \MT@scale\@tempcntb \@tempb \MT@dimen@six
829   \ifnum\@tempcntb=\z@ \else
830     \MT@scale@factor
831   \fi
832 }

```

`\MT@get@charwd` Get the width of the character. When using e-T<sub>E</sub>X, we can employ `\fontcharwd` instead of building scratch boxes.

```

833 \MT@requires@etex{
834   \def\MT@get@charwd{%
835     \MT@count=\fontcharwd\MT@font\MT@char\relax
836     \ifnum\MT@count=\z@ \MT@info@missing@char \fi
837   }
838 }{
839   \def\MT@get@charwd{%
840     \setbox\z@=\hbox{\MT@font \char\MT@char}%
841     \MT@count=\wd\z@\relax
842     \ifnum\MT@count=\z@ \MT@info@missing@char \fi
843   }
844 }

```

No adjustment with versions 0.14f and 0.14g.

```

845 }{
846   \def\MT@scale@to@em{%
847     \MT@count=\@tempb\relax
848     \ifnum\MT@count=\z@ \else
849       \MT@scale@factor
850     \fi
851   }

```

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

```

852   \def\MT@get@charwd{\MT@count=\MT@dimen@six}
853 }

```

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

```

854 \def\MT@get@font@dimen#1{%
855   \ifnum\fontdimen#1\MT@font=\z@
856     \MT@warning@nl{Font \font@name does not specify its\MessageBreak
857       \@backslashchar fontdimen \number#1 (it's zero).\MessageBreak
858       You should use a different `unit' for \MT@curr@list@name}%
859   \else
860     \MT@count=\fontdimen#1\MT@font
861   \fi
862 }

```

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

```

863 \MT@requires@etex{
864   \def\MT@info@missing@char{%
865     \MT@info@nl{Character `the\MT@toks' \iffontchar\MT@font\MT@char
866       has a width of 0pt \else is missing \fi in font\MessageBreak
867       `the\MT@exp@string\MT@font'. Ignoring protrusion settings\MessageBreak
868       for this character}%
869   }

```

```

870 }{
871   \def\MT@info@missing@char{%
872     \MT@info@n1{%
873       Character '\the\MT@toks' has a width of 0pt\MessageBreak
874       (it's probably missing) in font '\MT@exp@string\MT@font'.\MessageBreak
875       Ignoring protrusion settings for this character}%
876   }
877 }

```

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

```

878 \def\MT@scale@factor{%
879   \ifnum\csname MT@\MT@feat @factor\endcsname=\@m \else
880     \expandafter\MT@scale\expandafter
881     \@tempcntb \csname MT@\MT@feat @factor\endcsname \@m
882   \fi
883   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
884     \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @max\endcsname
885   \else
886     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
887       \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @min\endcsname
888     \fi
889   \fi
890 }

```

\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.

```

891 \def\MT@warn@code@too@large#1{%
892   \@tempcnta=#1\relax
893   \ifnum\csname MT@\MT@feat @factor\endcsname=\@m \else
894     \expandafter\MT@scale\expandafter\@tempcnta\expandafter\@m
895     \csname MT@\MT@feat @factor\endcsname
896   \fi
897   \MT@scale\@tempcnta \MT@dimen@six \MT@count
898   \MT@warning@n1{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
899     is too large for character\MessageBreak
900     '\the\MT@toks' in \MT@curr@list@name.\MessageBreak
901     Setting it to the maximum of \number\@tempcnta}%
902   \@tempcntb=#1\relax
903 }

```

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

```

904 \def\MT@get@opt{%

```

\MT@pr@factor@ Apply a factor?

```

\MT@sp@factor@ 905 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 906   \MT@let@nn{MT@\MT@feat @factor@}
907   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
908   \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
909     \number\csname MT@\MT@feat @factor\endcsname/1000}%
910   }{%
911     \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
912   }%

```

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

```

\MT@sp@unit@
\MT@kn@unit@ 913 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
914   \MT@let@nn{MT@\MT@feat @unit@}%
915   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%

```

```

916 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\empty
917 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
918         relative to character widths}%
919 \else
920 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\m@ne
921 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
922         relative to width of space}%
923 \fi
924 \fi
925 }{%
926 \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit@}%
927 }%

```

\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.

```

928 \let\MT@get@char@unit\relax
929 \let\MT@get@space@unit\@gobble
930 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\empty
931 \let\MT@get@char@unit\MT@get@charwd
932 \else
933 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\m@ne
934 \let\MT@get@space@unit\MT@get@font@dimen
935 \else
936 \expandafter\MT@get@unit\csname MT@\MT@feat @unit@\endcsname
937 \fi
938 \fi

```

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

```

939 \MT@ifdefined@nT{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
940 \MT@set@listname
941 \csname MT@preset@\MT@feat\endcsname
942 \MT@let@nc{MT@reset@\MT@feat @codes}\relax
943 }%
944 }

```

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

```

945 \def\MT@get@unit#1{%
946 \expandafter\MT@get@unit@#1 e!\@nil
947 \ifx\x\@empty\else\let#1\x\fi
948 \@defaultunits\@tempdima#1 pt\relax\@nnil
949 \ifdim\@tempdima=\z@
950 \MT@warning@n1{%
951 Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
952 width. Setting factors of list \@nameuse{MT@\MT@feat @c@name}'\MessageBreak
953 relative to character widths instead}%
954 \let#1\@empty
955 \let\MT@get@char@unit\MT@get@charwd
956 \else
957 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
958         to \the\@tempdima}%
959 \MT@count=\@tempdima\relax
960 \fi
961 }
962 \def\MT@get@unit@#1e#2#3\@nil{%
963 \ifx\#3\\\let\x\@empty \else
964 \if m#2%

```

```

965     \edef\x{#1\fontdimen6\MT@font}%
966   \else
967     \if x#2%
968       \edef\x{#1\fontdimen5\MT@font}%
969     \fi
970   \fi
971 \fi
972 }

```

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

```

973 \def\MT@set@pr@heirs#1{%
974   \lcode\MT@font#1=\lcode\MT@font\MT@char
975   \rcode\MT@font#1=\rcode\MT@font\MT@char
976   *debug
977   \MT@info@nl{2}{-- heir of \MT@char: #1}%
978   \MT@info@nl{4}{;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
979                                     \number\rcode\MT@font\MT@char}%
980   /debug
981 }

```

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

```

\MT@preset@pr@ 982 \def\MT@preset@pr{%
983   \expandafter\expandafter\expandafter\MT@preset@pr@
984   \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
985 }
986 \def\MT@preset@pr@#1,#2\@nil{%
987   \ifx\MT@pr@unit@\empty
988     \MT@warn@preset@twidth{pr}%
989     \let\MT@preset@aux\MT@preset@aux@factor
990   \else
991     \let\MT@preset@aux\MT@preset@aux@space
992   \fi
993   \MT@preset@aux{#1}\@tempa
994   \MT@preset@aux{#2}\@tempb
995   \MT@set@a11@pr\@tempa\@tempb
996 }

```

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

```

\MT@preset@aux@factor 997 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 998   \@tempcntb=#1\relax
999   \MT@scale@factor
1000   \edef#2{\number\@tempcntb}%
1001 }
1002 \def\MT@preset@aux@space#1#2{%
1003   \def\@tempb{#1}%
1004   \MT@get@space@unit\tw@
1005   \MT@scale@to@em
1006   \edef#2{\number\@tempcntb}%
1007 }

```

`\MT@warn@preset@twidth`

```

1008 \def\MT@warn@preset@twidth#1{%
1009   \MT@warning@nl{%
1010     Cannot preset characters relative to their widths\MessageBreak
1011     for \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1012     \MessageBreak relative to lem instead}%
1013 }

```



### 13.4.2 Expansion

`\MT@expansion` Set up for expansion?

```
1014 \def\MT@expansion{\MT@maybe@do{ex}}
```

`\MT@set@ex@codes@s` 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).

```
1015 \def\MT@set@ex@codes@s{%
1016   \MT@if@list@exists{%
1017     \MT@get@ex@opt
1018     \MT@reset@ef@codes
1019     \MT@get@inh@list
1020     \MT@load@list\MT@ex@cname
1021     \MT@set@listname
1022     \MT@let@cn\@tempc{\MT@ex@cname}%
1023     \expandafter\MT@ex@do\@tempc,\relax,%
1024     \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1025   }\relax
1026 }
```

`\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.

```
1027 \newif\ifMT@nonselected
1028 \def\MT@set@ex@codes@n{%
1029   \MT@nonselectedtrue
1030   \MT@if@list@exists
1031   \MT@get@ex@opt
1032   {%
1033     \let\MT@stretch@\MT@stretch
1034     \let\MT@shrink@\MT@shrink
1035     \let\MT@step@\MT@step
1036     \let\MT@auto@\MT@auto
1037     \let\MT@ex@factor@\MT@ex@factor
1038   }%
1039   \MT@reset@ef@codes
1040   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1041   \MT@nonselectedfalse
1042 }
```

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

```
1043 \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 the factor of this font).

`\MT@reset@ef@codes@`

```
1044 \def\MT@set@all@ex#1{%
1045   debug\MT@info{n}{3}{-- ex: setting all to \number{#1}}%
1046   \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1047 }
1048 \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.

```
1049 \MT@requires@pdftex4{
1050   \def\MT@reset@ef@codes{%
```

```

1051 \ifnum\MT@ex@factor@=\@m \else
1052 \MT@reset@ef@codes@
1053 \fi
1054 }
1055 }{
1056 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1057 }

```

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

```

1058 \def\MT@ex@do#1,{%
1059 \ifx\relax#1\@empty \else
1060 \MT@ex@split #1=\relax
1061 \expandafter\MT@ex@do
1062 \fi
1063 }

```

**\MT@ex@split**

```

1064 \def\MT@ex@split#1=#2=#3\relax{%
1065 \def\@tempa{#1}%
1066 \ifx\@tempa\@empty \else
1067 \MT@get@slot
1068 \ifnum\MT@char > \m@ne
1069 \@tempcntb=#2\relax

```

Take an optional factor into account.

```

1070 \ifnum\MT@ex@factor@=\@m \else
1071 \MT@scale\@tempcntb \MT@ex@factor@ \@m
1072 \fi
1073 \ifnum\@tempcntb > \MT@ex@max
1074 \MT@warn@ex@too@large\MT@ex@max
1075 \else
1076 \ifnum\@tempcntb < \MT@ex@min
1077 \MT@warn@ex@too@large\MT@ex@min
1078 \fi
1079 \fi
1080 \efcode\MT@font\MT@char=\@tempcntb
1081 (debug) \MT@dinfo@n1{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#2]}%

```

Heirs, heirs, I love thy heirs.

```

1082 \MT@ifdefined@c@T\MT@ex@inh@name{%
1083 \MT@ifdefined@n@T{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1084 \expandafter\MT@map@tlist@
1085 \csname MT@inh@\MT@ex@inh@name @\MT@char @\endcsname
1086 \MT@set@ex@heirs
1087 }%
1088 }%
1089 \fi
1090 \fi
1091 }

```

**\MT@warn@ex@too@large**

```

1092 \def\MT@warn@ex@too@large#1{%
1093 \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1094 character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1095 Setting it to the maximum of \number#1}%
1096 \@tempcntb=#1\relax
1097 }

```

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

```

\MT@ex@factor@ 1098 \def\MT@get@ex@opt{%
\MT@stretch@ 1099 \MT@ifdefined@n@TF{\MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@
\MT@step@
\MT@auto@

```

```

1100 \MT@let@cn\MT@ex@factor@{MT@ex@c@MT@ex@c@name @factor}%
1101 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
1102 }{%
1103 \let\MT@ex@factor@\MT@ex@factor
1104 }%
1105 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1106 \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1107 \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
1108 \def\@tempa{autoexpand}%
1109 \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1110 \MT@ifdefined@nT{MT@ex@c@\csname MT@ex@c@name\endcsname @preset}{%
1111 \MT@set@listname
1112 \MT@preset@ex
1113 \let\MT@reset@ef@codes\relax
1114 }%
1115 }

\MT@get@ex@opt@

1116 \def\MT@get@ex@opt@#1#2{%
1117 \MT@ifdefined@nTF{MT@ex@c@\MT@ex@c@name @#1}{%
1118 \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
1119 \MT@vinfo{... : #2}%
1120 }{%
1121 \MT@let@nn{MT@#1@}{MT@#1}%
1122 }%
1123 }

\MT@set@ex@heirs

1124 \def\MT@set@ex@heirs#1{%
1125 \efcode\MT@font#1=\efcode\MT@font\MT@char
1126 *debug
1127 \MT@dinfo@n1{2}{-- heir of \MT@char: #1}%
1128 \MT@dinfo@n1{4}{:: ef (#1) \number\efcode\MT@font\MT@char}%
1129 /debug
1130 }

\MT@preset@ex

1131 \def\MT@preset@ex{%
1132 \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
1133 \MT@scale@factor
1134 \MT@set@all@ex@\@tempcntb
1135 }
1136 *beta

```

### 13.4.3 Interword Space (Glue)

`\MT@spacing` Adjustment of interword spacing? Only for sufficiently new versions of pdfTeX.

```

1137 \MT@requires@pdftex6{
1138 \def\MT@spacing{\MT@maybe@do{sp}}
1139 }{
1140 \let\MT@spacing\relax
1141 }

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

1142 \def\MT@set@sp@codes{%
1143 \MT@if@list@exists{%
1144 \MT@get@dimen@six
1145 \MT@get@opt
1146 \MT@reset@sp@codes

```

```

1147 \MT@get@inh@list
1148 \MT@load@list\MT@sp@c@name
1149 \MT@set@listname
1150 \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
1151 \expandafter\MT@sp@do\@tempc,\relax,%
1152 }\MT@reset@sp@codes
1153 }

\MT@sp@do
1154 \def\MT@sp@do#1,{%
1155 \ifx\relax#1\@empty \else
1156 \MT@sp@split #1=\relax
1157 \expandafter\MT@sp@do
1158 \fi
1159 }

\MT@sp@split
1160 \def\MT@sp@split#1=#2=#3\relax{%
1161 \def\@tempa{#1}%
1162 \ifx\@tempa\@empty \else
1163 \MT@get@slot
1164 \ifnum\MT@char > \m@ne
1165 \MT@get@char@unit
1166 \MT@sp@split@val#2\relax
1167 \fi
1168 \fi
1169 }

\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).
1170 \def\MT@sp@split@val#1,#2,#3\relax{%
1171 \def\@tempb{#1}%
1172 \MT@ifempty\@tempb\relax{%
1173 \MT@get@space@unit\tw@
1174 \MT@scale@to@em
1175 \knbscode\MT@font\MT@char=\@tempcntb
1176 (debug)\MT@info@n1{4}{;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1177 }%
1178 \def\@tempb{#2}%
1179 \MT@ifempty\@tempb\relax{%
1180 \MT@get@space@unit\thr@@
1181 \MT@scale@to@em
1182 \stbscode\MT@font\MT@char=\@tempcntb
1183 (debug)\MT@info@n1{4}{;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1184 }%
1185 \def\@tempb{#3}%
1186 \MT@ifempty\@tempb\relax{%
1187 \MT@get@space@unit4%
1188 \MT@scale@to@em
1189 \shbscode\MT@font\MT@char=\@tempcntb
1190 (debug)\MT@info@n1{4}{;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1191 }%
1192 \MT@ifdefined@c@T\MT@sp@inh@name{%
1193 \MT@ifdefined@n@T{MT@inh@\MT@sp@inh@name @\MT@char @}{%
1194 \expandafter\MT@map@tlist@c
1195 \csname MT@inh@\MT@sp@inh@name @\MT@char @\endcsname
1196 \MT@set@sp@heirs
1197 }%
1198 }%
1199 }

```

```

\MT@set@sp@heirs
1200 \def\MT@set@sp@heirs#1{%
1201   \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1202   \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1203   \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1204   *debug
1205   \MT@info@nl{2}{-- heir of \MT@char: #1}%
1206   \MT@info@nl{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1207                                           \number\stbscode\MT@font\MT@char/%
1208                                           \number\shbscode\MT@font\MT@char}%
1209   /debug
1210 }

\MT@set@all@sp
\MT@reset@sp@codes 1211 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1212 debug\MT@info@nl{3}{-- knbs/stbs/shbs: setting all to \number#1/\number#2/\number#3}%
1213   \MT@do@font{%
1214     \knbscode\MT@font\@tempcnta=#1\relax
1215     \stbscode\MT@font\@tempcnta=#2\relax
1216     \shbscode\MT@font\@tempcnta=#3\relax
1217   }%
1218 }
1219 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1220 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1221 \def\MT@preset@sp{%
1222   \expandafter\expandafter\expandafter\MT@preset@sp@
1223   \cscname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1224 }
1225 \def\MT@preset@sp@#1,#2,#3\@nil{%
1226   \ifx\MT@sp@unit@\empty
1227     \MT@warn@preset@twidth{sp}%
1228     \MT@preset@aux@factor{#1}\@tempa
1229     \MT@preset@aux@factor{#2}\@tempc
1230     \MT@preset@aux@factor{#3}\@tempb
1231   \else
1232     \MT@preset@aux@space{#1}\@tempa
1233     \def\@tempb{#2}%
1234     \MT@get@space@unit\thr@@
1235     \MT@scale@to@em
1236     \edef\@tempc{\number\@tempcntb}%
1237     \def\@tempb{#3}%
1238     \MT@get@space@unit4%
1239     \MT@scale@to@em
1240     \edef\@tempb{\number\@tempcntb}%
1241   \fi
1242   \MT@set@all@sp\@tempa\@tempc\@tempb
1243 }

```

#### 13.4.4 Additional Kerning

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

```

1244 \MT@requires@pdftex6{
1245   \def\MT@kerning{\MT@maybe@do{kn}}
1246 }{
1247   \let\MT@kerning\relax
1248 }

```

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

```

1249 \def\MT@set@kn@codes{%
1250   \MT@if@list@exists{%
1251     \MT@get@dimen@six
1252     \MT@get@opt
1253     \MT@reset@kn@codes
1254     \MT@get@inh@list
1255     \MT@load@list\MT@kn@c@name
1256     \MT@set@listname
1257     \MT@let@cn\@tempc{\MT@kn@c@\MT@kn@c@name}%
1258     \expandafter\MT@kn@do\@tempc,\relax,%
1259   }\MT@reset@kn@codes
1260 }
```

\MT@kn@do

```

1261 \def\MT@kn@do#1,{%
1262   \ifx\relax#1\@empty \else
1263     \MT@kn@split #1=\relax
1264     \expandafter\MT@kn@do
1265   \fi
1266 }
```

\MT@kn@split

```

1267 \def\MT@kn@split#1=#2=#3\relax{%
1268   \def\@tempa{#1}%
1269   \ifx\@tempa\@empty \else
1270     \MT@get@slot
1271     \ifnum\MT@char > \m@ne
1272       \MT@get@char@unit
1273       \MT@kn@split@val#2\relax
1274     \fi
1275   \fi
1276 }
```

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

```

1277 \def\MT@kn@split@val#1,#2\relax{%
1278   \def\@tempb{#1}%
1279   \MT@ifempty\@tempb\relax{%
1280     \MT@get@space@unit\tw@
1281     \MT@scale@to@em
1282     \knbcode\MT@font\MT@char=\@tempcntb
1283     <debug>\MT@info@n1{4}{;;; knbc (\MT@char): \number\knbcode\MT@font\MT@char: [#1]}%
1284   }%
1285   \def\@tempb{#2}%
1286   \MT@ifempty\@tempb\relax{%
1287     \MT@get@space@unit\tw@
1288     \MT@scale@to@em
1289     \knaccode\MT@font\MT@char=\@tempcntb
1290     <debug>\MT@info@n1{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1291   }%
1292   \MT@ifdefined@c@T\MT@kn@inh@name{%
1293     \MT@ifdefined@n@T\MT@inh@\MT@kn@inh@name @\MT@char @}%
1294     \expandafter\MT@map@tlist@c
1295     \csname MT@inh@\MT@kn@inh@name @\MT@char @\endcsname
1296     \MT@set@kn@heirs
1297   }%
1298 }%
1299 }
```

\MT@set@kn@heirs

```

1300 \def\MT@set@kn@heirs#1{%
1301   \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1302   \knacode\MT@font#1=\knacode\MT@font\MT@char
1303   *debug
1304   \MT@info{n}{2}{-- heir of \MT@char: #1}%
1305   \MT@info{n}{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1306                                     \number\knacode\MT@font\MT@char}%
1307   /debug
1308 }

\MT@set@all@kn
\MT@reset@kn@codes 1309 \def\MT@set@all@kn#1#2{%
1310   debug \MT@info{n}{3}{-- knac/knbc: setting all to \number#1/\number#2}%
\MT@reset@kn@codes@ 1311   \MT@do@font{%
1312     \knbcode\MT@font\@tempcnta=#1\relax
1313     \knacode\MT@font\@tempcnta=#2\relax
1314   }%
1315 }
1316 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1317 \let\MT@reset@kn@codes\relax

\MT@preset@kn
\MT@preset@kn@ 1318 \def\MT@preset@kn{%
1319   \expandafter\expandafter\expandafter\MT@preset@kn@
1320   \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1321 }
1322 \def\MT@preset@kn@#1,#2\@nil{%
1323   \ifx\MT@kn@unit@\@empty
1324     \MT@warn@preset@tewidth{kn}%
1325     \let\MT@preset@aux\MT@preset@aux@factor
1326   \else
1327     \let\MT@preset@aux\MT@preset@aux@space
1328   \fi

Letterspacing factor.
1329   \MT@ifstreq\MT@kn@context{letterspacing}{%
1330     \@tempcnta\MT@kn@factor@\relax
1331     \MT@scale\@tempcnta \MT@letterspacing@ \@m
1332     \edef\MT@kn@factor@{\number\@tempcnta}%
1333   }\relax
1334   \MT@preset@aux{#1}\@tempa
1335   \MT@preset@aux{#2}\@tempb
1336   \MT@set@all@kn\@tempa\@tempb
1337 }

```

### 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.

```

1338 \MT@requires@pdftex6{
1339   \DeclareRobustCommand\lsstyle{%
1340     \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`.

```

1341     \begingroup
1342     \escapechar\m@ne
1343     \expandafter\MT@exp@two@n\expandafter\MT@in@clist\expandafter
1344     {\expandafter\string\expandafter

```

```

1345         \csname\curr@fontshape/\f@size\expandafter\endcsname}%
1346         {\csname MT@knlist@font@MT@kn@setname\endcsname}%
1347         \ifMT@inlist@ \else
1348         \expandafter\MT@xadd
1349         \csname MT@knlist@font@MT@kn@setname\endcsname
1350         {\MT@exp@string\csname\curr@fontshape/\f@size\endcsname,}%
1351         \fi
1352     \endgroup
1353 \fi
1354 \MT@kerningtrue
1355 \pdfappendkern\@ne
1356 \pdfprependkern\@ne
1357 \def\MT@kn@setname{all}%
1358 \MT@ifdefined@c@TF\MT@letterspacing@relax{%
1359     \let\MT@letterspacing@\MT@letterspacing
1360 }%
1361 \microtypecontext{kerning=letterspacing}%
1362 }
1363 }{
1364 \DeclareRobustCommand\lsstyle{%
1365     \MT@warning{Letterspacing only works with pdftex version 1.40\MessageBreak
1366     or newer. You might want to use the `soul' package\MessageBreak
1367     instead}%
1368     \global\let\lsstyle\relax
1369 }
1370 }

```

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

```

1371 \DeclareRobustCommand\textls[2][ ]{%
1372     \MT@ifempty{#1}{%
1373         \let\MT@letterspacing@\undefined
1374     }{%
1375         \KV@esp@def\MT@letterspacing@{#1 }%
1376     }%
1377     {\lsstyle #2}%
1378 }
1379 /beta

```

### 13.4.6 Disabling Ligatures

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

```

1380 \MT@requires@pdftex5{
1381     \def\MT@noligatures{%
1382         \ifMT@noligatures
1383             \MT@dotrue
1384             \MT@map@clist{n{font,encoding,family,series,shape,size}}{%
1385                 \MT@ifdefined@n@TF{MT@checklist@##1}%
1386                 {\csname MT@checklist@##1\endcsname}%
1387                 {\MT@checklist@{##1}}%
1388             }{nl}%
1389         }%
1390     \else
1391         \MT@dofalse
1392     \fi
1393     \ifMT@do
1394         \pdfnoligatures\MT@font
1395         \MT@vinfo{... Disabling ligatures}%
1396     \fi
1397 }

```



```

1398 }{
1399   \let\MT@noligatures\relax
1400 }

```

### 13.4.7 Loading the Configuration

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

```

1401 \def\MT@load@list#1{%
1402   \edef\@tempa{#1}%
1403   \MT@let@cn\@tempb{\MT@MT@feat @c@\@tempa load}%
1404   \MT@ifstreq\@tempa\@tempb{%
1405     \MT@warning{\@nameuse{\MT@abbr@\MT@feat} list '\@tempa' cannot load itself}%
1406   }{%
1407     \ifx\@tempb\relax \else
1408       \MT@ifdefined@n@TF{\MT@\MT@feat @c@\@tempb}{%
1409         \MT@vinfo{... : First loading \@nameuse{\MT@abbr@\MT@feat} list '\@tempb'}%
1410         \begingroup
1411           \MT@load@list\@tempb
1412         \endgroup
1413         \edef\MT@curr@list@name{\@nameuse{\MT@abbr@\MT@feat} list\MessageBreak'\@tempb'}%
1414         \MT@let@cn\@tempc{\MT@\MT@feat @c@\@tempb}%
1415         \expandafter\csname \MT@\MT@feat @do\expandafter\endcsname\@tempc,\relax,%
1416       }{%
1417         \MT@warning{\@nameuse{\MT@abbr@\MT@feat} list '\@tempb' undefined.
1418           Cannot load\MessageBreak it from list '\@tempa'}%
1419       }%
1420     \fi
1421   }%
1422 }

```

\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.

```

1423 \let\MT@file@list\@empty
1424 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

1425   \MT@in@clist{#1}\MT@file@list
1426   \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.

```

1427   \MT@begin@catcodes
1428   \let\MT@begin@catcodes\relax
1429   \let\MT@end@catcodes\relax
1430   \InputIfFileExists{mt-#1.cfg}{%
1431     \edef\MT@curr@file{mt-#1.cfg}%
1432     \MT@vinfo{... Loading configuration file \MT@curr@file}%
1433     \MT@xadd\MT@file@list{#1,%
1434   }{%
1435     \expandafter\MT@get@basefamily#1\relax\relax\relax
1436     \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1437     \ifMT@inlist\else
1438       \InputIfFileExists{mt-\@tempa.cfg}{%
1439         \edef\MT@curr@file{mt-\@tempa.cfg}%
1440         \MT@vinfo{... Loading configuration file \MT@curr@file}%
1441         \MT@xadd\MT@file@list{\@tempa,#1,%
1442       }{%
1443         \MT@vinfo{... No configuration file mt-#1.cfg}%
1444         \MT@xadd\MT@file@list{#1,%

```

```

1445         }%
1446     \fi
1447     }%
1448 \endgroup
1449 \fi
1450 }

```

`\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), !, ?, , , \$, \_ , ~, and = (Turkish babel).

OK, now all printable characters up to 127 are 'other'. We hope that letters are always letters and numbers other. (`\listings` makes them active, see section 13.3.)

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.

```

1451 \def\MT@begin@catcodes{%
1452     \begingroup
1453     \makeatletter
1454     \catcode`\^7%
1455     \catcode`\_9%
1456     \catcode`\^^I9%
1457     \catcode`\^^M9%
1458     \catcode`\z0
1459     \catcode`\{@ne
1460     \catcode`\}\tw@
1461     \catcode`\#6%
1462     \catcode`\%14%
1463     \MT@map@tlist@n
1464     {\!\\"\$&\'\(\)\*\+\,\-\./\:\;\<=\>\?[\]\_-\`|\~}%
1465     \@makeoother

```

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

```

1466 \let\KV@@sp@def\def
1467 }

```

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

```

1468 \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 ...).

```

1469 \def\MT@get@basefamily#1#2#3#4\relax{%
1470     \ifx#2\relax \def\@tempa{#1}\else
1471         \ifx#3\relax \def\@tempa{#1#2}\else
1472             \def\@tempa{#1#2#3}%
1473         \ifx\relax#4\relax \else
1474             \MT@ifstreq{#4}{\string x}\relax{%
1475                 \MT@ifstreq{#4}{\string j}\relax{%
1476                     \MT@ifstreq{#4}{\string w}\relax{%
1477                         \def\@tempa{#1#2#3#4}}}\fi\fi\fi
1478 }

```

`\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@ 1479 \def\MT@get@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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```

1480 <debug>\MT@info@n1{1}{trying to find \@nameuse{MT@abbr@#1} list for font \MT@font}%
1481 \let\MT@listname\@undefined
1482 \def\@tempb{#1}%
1483 \MT@map@tlist@c\MT@try@order\MT@get@listname@
1484 }
1485 \def\MT@get@listname@#1{%
1486 \expandafter\MT@next@listname#1%
1487 \ifx\MT@listname\@undefined \else
1488 \expandafter\MT@tlist@break
1489 \fi
1490 }

```

\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.

```

1491 \def\MT@try@order{%
1492 {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
1493 {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
1494 }

```

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

```

1495 \def\MT@next@listname#1#2#3#4{%
1496 \edef\@tempa{\MT@encoding
1497 \ifnum#1=\@ne \MT@family\fi
1498 \ifnum#2=\@ne \MT@series\fi
1499 \ifnum#3=\@ne \MT@shape\fi
1500 \ifnum#4=\@ne *\fi
1501 \MT@context}%
1502 <debug>\MT@info@n1{1}{trying \@tempa}%
1503 \MT@ifdefined@n@TF{MT@\@tempb \@tempa}{%
1504 \MT@next@listname@#4%
1505 }{%

```

Also try with an alias family.

```

1506 \ifnum#1=\@ne
1507 \ifx\MT@familyalias\@empty \else
1508 \edef\@tempa{\MT@encoding
1509 \MT@familyalias
1510 \ifnum#2=\@ne \MT@series\fi
1511 \ifnum#3=\@ne \MT@shape\fi
1512 \ifnum#4=\@ne *\fi
1513 \MT@context}%
1514 <debug>\MT@info@n1{1}{(alias) \@tempa}%
1515 \MT@ifdefined@n@TF{MT@\@tempb \@tempa}{%
1516 \MT@next@listname@#4%
1517 }%
1518 \fi
1519 \fi
1520 }%

```

```

1521 }
\MT@next@listname@ If size is to be evaluated, do that, otherwise use the current list.
1522 \def\MT@next@listname@#1{%
1523   \ifnum#1=\@ne
1524     \expandafter\MT@in@rlist\csname MT@\@tempb @\@tempa @sizes\endcsname
1525     \ifMT@inlist@
1526       \let\MT@listname\MT@size@name
1527     \fi
1528   \else
1529     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
1530   \fi
1531 }

```

\MT@if@list@exists

```

\MT@context 1532 \def\MT@if@list@exists{%
1533   \MT@let@cn\MT@context{MT@MT@feat @context}%
1534   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
1535   \MT@get@listname{\MT@feat @c}%
1536   \MT@ifdefined@c@TF\MT@listname{%
1537     \MT@edef@n{MT@MT@feat @c@name}{\MT@listname}%
1538     \ifMT@nonselected
1539       \MT@vinfo{... Applying non-selected expansion (list `~\MT@ex@c@name')}%
1540     \else
1541       \MT@vinfo{... Loading \@nameuse{MT@abbr@MT@feat} list
1542         ~\@nameuse{MT@MT@feat @c@name}'}%
1543     \fi
1544     \@firstoftwo
1545   }{%

```

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

```

1546   \MT@let@nc{MT@MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

1547   \ifMT@nonselected
1548     \MT@vinfo{... Applying non-selected expansion}%
1549   \else
1550     \MT@warning{I cannot find a \@nameuse{MT@abbr@MT@feat} list
1551       for font\MessageBreak~\MT@exp@string\MT@font'%
1552       \ifx\MT@context\@empty\else\space(context: ~\MT@context')\fi.
1553       Switching off\MessageBreak\@nameuse{MT@abbr@MT@feat} for this font}%
1554     \fi
1555     \@secondoftwo
1556   }%
1557 }

```

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

```

\MT@context 1558 \def\MT@get@inh@list{%
1559   \let\MT@context\@empty
1560   \MT@get@listname{\MT@feat @inh}%
1561   \MT@ifdefined@c@TF\MT@listname{%
1562     \MT@edef@n{MT@MT@feat @inh@name}{\MT@listname}%
1563     *debug
1564     \MT@dinfo@n1{1}{... Using \@nameuse{MT@abbr@MT@feat} inheritance list
1565       ~\@nameuse{MT@MT@feat @inh@name}'}%
1566     /debug
1567     \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.

```

1568   \ifx\@tempc\@empty \else

```

```

1569 <debug>\MT@info@n1{1}{parsing inheritance list ...}%
1570 \MT@let@cn\MT@inh@name{MT@\MT@feat @inh@name}%
1571 \edef\MT@curr@list@name{inheritance list\MessageBreak`~\MT@inh@name'}%
1572 \expandafter\MT@inh@do\@tempc,\relax,%
1573 \global\MT@let@nc{MT@\MT@feat @inh@\csname MT@\MT@feat @inh@name\endcsname}\@empty
1574 \fi
1575 }{}%
1576 \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
1577 }%
1578 }

```

### 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.

```

1579 \def\MT@get@slot{%
1580 \escapechar~\
1581 \let\MT@char@m@ne
1582 \MT@noresttrue

```

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

```

1583 \MT@toks=\expandafter{\@tempa}%

```

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

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

```

1584 \expandafter\MT@is@letter\@tempa\relax\relax
1585 \ifnum\MT@char@ < \z@

```

- It might be an active character, i. e., an 8-bit character defined by inputenc. If so, we will expand it here to its LICR form.

```

1586 \MT@exp@two@c\MT@is@active\string\@tempa\@nil

```

- OK, so it must be a macro. We do not allow random commands but only those defined in L<sup>A</sup>T<sub>E</sub>X's idiosyncratic font encoding scheme:

If  $\langle encoding \rangle \langle command \rangle$  (that's *one* command) is defined, we try to extract the slot number.

We must be cautious not to stumble over accented characters consisting of two commands, like  $\backslash'i$  or  $\backslash U\backslash CYRI$ , hence,  $\backslash string$  wouldn't be safe enough.

```

1587 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
1588 \MT@is@symbol

```

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

```

1589 {\expandafter\MT@is@composite\@tempa\relax\relax}%
1590 \ifnum\MT@char@ < \z@

```

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

```

1591      \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1592      \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
1593      \fi
1594  \fi

1595  \let\MT@char\MT@char@
1596  \ifnum\MT@char < \z@
1597    \MT@warn@unknown
1598  \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.

```

1599  \ifMT@noreset \else
1600    \MT@warn@reset
1601    \let\MT@char\m@ne
1602  \fi
1603  \fi
1604  \escapechar\m@ne
1605 }

```

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

```

\MT@testrest 1606 \newif\ifMT@noreset
1607 \def\MT@testrest#1#2{\MT@ifstreq{#1}{#2}\relax\MT@noresetfalse}

```

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

```

1608 \def\MT@is@letter#1#2\relax{%
1609   \ifcat a\noexpand#1\relax
1610     \edef\MT@char@{\number`#1}%
1611     \ifx\#2\%
1612 <debug> \MT@dinfo@n1{3}{> `the\MT@toks' is a letter (\MT@char@)}%
1613     \else
1614       \MT@noresetfalse
1615     \fi
1616   \else
1617     \ifcat !\noexpand#1\relax
1618       \edef\MT@char@{\number`#1}%
1619 <debug> \MT@dinfo@n1{3}{> `the\MT@toks' is a character (\MT@char@)}%
1620       \ifx\#2\%
1621         \ifnum\MT@char@>127 \MT@warn@ascii \fi
1622       \else
1623         \MT@noresetfalse
1624         \expandafter\MT@is@number#1#2\relax\relax
1625       \fi
1626     \fi
1627   \fi
1628 }

```

`\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.

```

1629 \def\MT@is@number#1#2#3\relax{%
1630   \ifx\relax#3\relax \else
1631     \ifx\relax#2\relax \else
1632       \MT@noresettrue
1633       \if#1"\relax
1634         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
1635 <debug> \MT@dinfo@n1{3}{> ... a hexadecimal number: \MT@char@}%
1636       \else

```

```

1637      \if#1'\relax
1638      \def\MT@char@{\number#1#2#3}%
1639 (debug)\MT@info@n1{3}{> ... an octal number: \MT@char@}%
1640      \else
1641      \MT@ifint{#1#2#3}{%
1642      \def\MT@char@{\number#1#2#3}%
1643 (debug)\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
1644      }\MT@noestfalse
1645      \fi
1646      \fi
1647      \ifnum\MT@char@ > \@cclv
1648      \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
1649      \let\MT@char@\m@ne
1650      \fi
1651      \fi
1652      \fi
1653 }

```

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

Previous solution, slightly more robust (but doesn't understand Unicode):

```

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

Unicode characters (`inputenc/utf8`) are also supported (however, not `ucs`, aka.

inputenc/utf8x).

We presume that only one input encoding is being used throughout the entire document. We could of course save the input encoding together with the list, but this would entail a couple of problems: (1) the overhead of resetting the input encoding every time, (2) the problem (or rather, impossibility) to decide which input encoding should be chosen – the one active when the list was declared or the one active when the font is selected, and (3) the improbability of multiple input encodings being used at all.

```
1654 \def\MT@is@active#1#2\@nil{%
1655   \ifnum\catcode`#1 = \active
1656     \begingroup
1657       \set@display@protect
1658       \let\IeC\@firstofone
1659       \let\@inpenc\undefined\MT@undefined@char
```

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

```
1660   \def\UTFviii@defined##1{\ifx ##1\relax
1661     \MT@undefined@char{utf8}\else\expandafter##1\fi}%
1662   \edef\x{%
1663     \def\noexpand\@tempa{\@tempa}%
```

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

```
1664     \MT@toks={\the\MT@toks\space(= \@tempa)}%
1665   }%
1666   \expandafter\endgroup\x
1667 \fi
1668 }
```

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

```
1669 \def\MT@undefined@char#1{undefined in input encoding ``#1''}
```

\MT@is@symbol The symbol commands might expand to funny stuff, depending on context. Instead of simply expanding `\langle command \rangle`, we construct the command `\langle encoding \rangle \langle command \rangle` 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.

```
1670 \def\MT@is@symbol{%
1671   \edef\MT@char{\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
1672   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1673     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
1674   \ifnum\MT@char@ < \z@
```

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

```
1675   \expandafter\MT@is@letter\MT@char\relax\relax
1676 \fi
1677 }
```

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

```
\MT@charstring 1678 \begingroup
1679   \catcode`\/=0
1680   /MT@map@tlist@n{/\/C/H/A/R}/@makeother
1681   /lowercase{%
1682     /def/x{%
1683       /def\MT@charstring{\CHAR"%
1684       /def\MT@is@char##1\CHAR"##2##3##4/relax{%
1685         /ifx/relax##1/relax
```



```

1686         /if##3\relax
1687         /edef\MT@char@{/number"##2}%
1688         /MT@testrest/MT@charstring{##3##4}%
1689         /else
1690         /edef\MT@char@{/number"##2##3}%
1691         /MT@testrest/MT@charstring{##4}%
1692         /fi
1693 (debug) /MT@info@n1{3}{> `the\MT@toks' is a \char (/MT@char@)}%
1694         /fi
1695     }%
1696 }%
1697 }
1698 /expandafter/endgroup/x

```

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

```

1699 \def\MT@is@composite#1#2\relax{%
1700     \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, especially, since we also `\detokenize` the input instead of only `\stringifying` it. Thus, we will die gracefully even on wrong Unicode input without `utf8`.

```

1701     \edef\MT@char{\expandafter
1702         \csname\expandafter
1703             \string\csname\MT@encoding\endcsname
1704             \MT@detokenize@n{#1}-%
1705             \MT@detokenize@n{#2}%
1706             \endcsname}%
1707     \expandafter\MT@is@letter\MT@char\relax\relax
1708     \fi
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.

```

\MT@set@listname 1710 \def\MT@set@listname{%
1711     \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\MessageBreak
1712     \@nameuse{MT@\MT@feat @c@name}'}%
1713 }

```

`\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.

```

1714 \def\MT@warn@ascii{%
1715     \MT@warning@n1{Character `the\MT@toks' (= \MT@char@)
1716     is outside of ASCII range.\MessageBreak
1717     You must load the `inputenc' package before using\MessageBreak
1718     8-bit characters in \MT@curr@list@name}%
1719 }

```

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

```

1720 \def\MT@warn@number@too@large#1{%
1721     \MT@warning@n1{%
1722         Number #1 in encoding '\MT@encoding' too large!\MessageBreak
1723         Ignoring it in \MT@curr@list@name}%
1724 }

```

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

```

1725 \def\MT@warn@rest{%
1726   \MT@warning@nl{%
1727     Unknown slot number of character\MessageBreak`the\MT@toks'\MessageBreak
1728     in font encoding `\'MT@encoding'. Make sure it's a single\MessageBreak
1729     character (or a number) in \MT@curr@list@name}%
1730 }

```

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

```

1731 \def\MT@warn@unknown{%
1732   \MT@warning@nl{%
1733     Unknown slot number of character\MessageBreak`the\MT@toks'\MessageBreak
1734     in font encoding `\'MT@encoding' in \MT@curr@list@name}%
1735 }

```

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

For my reference:

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

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

```
1736 \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.

```
1737 \let\MT@font\empty
1738 \MT@addto@setup{%
1739   \g@addto@macro\do@subst@correction{%
1740     \xdef\MT@font{\csname \curr@fontshape/\f@size\endcsname}%
1741   }
```

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

```
1742 \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
1743 \ifx\pickup@font\MT@orig@pickupfont \else
1744   \MT@warning@nl{%
1745     Command \string\pickup@font space is not defined as expected.\MessageBreak
1746     Double-check whether micro-typography is indeed\MessageBreak
1747     applied to the document.\MessageBreak (Hint: Turn on `verbose' mode)%
1748   }
1749 \fi
```

Then we append our stuff.

```
1750 \g@addto@macro\pickup@font{%
1751   \begingroup
1752   \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.

```
1753   \ifx\MT@font\empty
1754     \let\MT@font\font@name
1755   \else
1756     \ifx\MT@font\font@name \else
1757       \MT@register@subst@font
1758     \fi
1759   \fi
1760   \MT@check@font
1761   \ifMT@inlist@ \else
1762     \MT@setupfont
1763     \MT@register@font
1764   \fi
1765 \endgroup
1766 \global\let\MT@font\empty
1767 }
```

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

```
1768 \let\MT@pickupfont\pickup@font
```

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

```
1769 \let\MT@orig@add@accent\add@accent
1770 \def\add@accent#1#2{%
1771   \let\pickup@font\MT@orig@pickupfont
1772   \MT@orig@add@accent{#1}{#2}%
1773   \let\pickup@font\MT@pickupfont
1774 }
1775 }
```

Consequently, we are the last one to change these commands, therefore there is no need to check whether our definition has survived.<sup>14</sup>

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

```
1776 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

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

```
1777 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
```

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

```
1778 \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.

```
1779 \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.

```
1780 \def\MT@check@font@cx{%
1781   \@tempswatrue
1782   \MT@map@clist@c\MT@active@features{%
1783     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
1784     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1785     \ifMT@inlist@
1786       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
1787     \else
1788       \@tempswafalse
1789     \fi
1790   }%
1791   \if@tempswa \MT@inlist@true \else \MT@inlist@false \fi
1792 }
```

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

```
1793 \def\MT@register@subst@font@cx{%
1794   \MT@map@clist@c\MT@active@features{%
1795     \expandafter\MT@xadd
1796     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1797     {\font@name,}%
1798   }%
1799 }
```

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

```
1800 \def\MT@register@font@cx{%
```

<sup>14</sup> If CJK is loaded, microtype will overwrite its definition. Sorry, Asia!

```

1801 \MT@map@clist@c\MT@active@features{%
1802 \expandafter\ifx\csname MT@\@nameuse{MT@abbr@##1}\endcsname\relax\else
1803 \expandafter\MT@xadd
1804 \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
1805 {\MT@font,}%
1806 \def\@tempa{##1}%
1807 \expandafter\MT@map@tlist@c
1808 \csname MT@##1@doc@contexts\endcsname
1809 \MT@rem@from@lists
1810 \fi
1811 }%
1812 }

```

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

```

1813 \def\MT@rem@from@lists#1{%
1814 \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
1815 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist
1816 \expandafter\MT@font\csname MT@\@tempa @#1font@list\endcsname
1817 }%
1818 }

```

`\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.

```

1819 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
1820 \MT@addto@setup{%
1821 \def\microtypecontext#1{%
1822 \MT@setup@contexts
1823 \setkeys{MTC}{#1}%
1824 \selectfont
1825 \aftergroup\MT@reset@context
1826 }%
1827 }

```

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

```

1828 \def\MT@reset@context{%
1829 \MT@vinfo{Resetting contexts on line \the\inputlineno
1830 <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
1831 <debug&beta> / \MT@kn@context/\MT@sp@context
1832 }%
1833 \selectfont
1834 }

```

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

```

1835 \def\MT@setup@contexts{%
1836 \MT@map@clist@c\MT@active@features
1837 {\global\MT@let@nc{MT@##1@font@list}\MT@font@list}%
1838 \global\let\MT@check@font\MT@check@font@cx
1839 \global\let\MT@register@font\MT@register@font@cx
1840 \global\let\MT@register@subst@font\MT@register@subst@font@cx
1841 \global\let\MT@setup@contexts\relax
1842 }

```

`\MT@define@context`

```

1843 \def\MT@define@context#1{%
1844 \define@key{MTC}{#1}[]{}%
1845 \KV@sp@def\@tempb{#1}%
1846 \edef\@tempb{\@nameuse{MT@rbba@\@tempb}}%

```

```

1847 \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
1848 \ifMT@inlist@

```

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

```

1849 \MT@ifempty{##1}{\def\MT@val{}}{\KV@sp@def\MT@val{##1}}%
1850 \expandafter\ifx\csname MT@\@tempb @context\endcsname\MT@val \else
1851 \MT@vinfo{--- Changing #1 context to '\MT@val'}%

```

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

```

1852 \global\MT@let@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes}%

```

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

```

1853 \expandafter\MT@exp@one@n\expandafter\MT@in@tlist
1854 \expandafter\MT@val\csname MT@\@tempb @doc@contexts\endcsname
1855 \ifMT@inlist@ \else
1856 \expandafter\MT@xadd\csname MT@\@tempb @doc@contexts\endcsname{\MT@val}%
1857 (debug) \MT@dinfo{2}{>>> added #1 context: \nameuse{MT@\@tempb @doc@contexts}}%
1858 \fi
1859 \MT@edef@n{MT@\@tempb @context}{\MT@val}%
1860 \fi
1861 \fi
1862 }%
1863 }
1864 \MT@map@clist@c\MT@features@long{\MT@define@context{#1}}

```

\MT@pr@context Initialize the contexts.

\MT@ex@context 1865 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%}

\MT@sp@context 1866 \MT@def@n{MT@#1@context}{@}%

\MT@kn@context 1867 \MT@def@n{MT@#1@doc@contexts}{@}%

1868 }

\MT@pr@doc@contexts 1869 \let\MT@extra@context\empty

\MT@ex@doc@contexts

\MT@sp@doc@contexts

\MT@kn@doc@contexts

\MT@extra@context

## 13.5 Configuration

### 13.5.1 Font Sets

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

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

```

1870 \def\DeclareMicrotypeSet{%
1871 \ifstar
1872 {\ifnextchar[\MT@DeclareSetAndUseIt
1873 {\MT@DeclareSetAndUseIt[]}}%
1874 {\ifnextchar[\MT@DeclareSet
1875 {\MT@DeclareSet[]}}%
1876 }

```

\MT@DeclareSet

\MT@DeclareSetAndUseIt 1877 \def\MT@DeclareSet[#1]{%

1878 \MT@DeclareSet@{#1}%

1879 }

1880 \def\MT@DeclareSetAndUseIt[#1]#2#3{%

1881 \MT@DeclareSet@{#1}{#2}{#3}%

1882 \UseMicrotypeSet[#1]{#2}%

```

1883 }

\MT@DeclareSet@
1884 \def\MT@DeclareSet@#1#2#3{%
1885   \KV@sp@def\@tempa{#1}%
1886   \MT@ifempty\@tempa{%
1887     \MT@map@clist@c\MT@features{\MT@declare@sets{##1}{#2}{#3}}}%
1888   }{%
1889     \MT@map@clist@c\@tempa{%
1890       \KV@sp@def\@tempa{##1}%
1891       \MT@ifempty\@tempa\relax{%
1892         \MT@is@feature{set declaration `#2'}{%
1893           \MT@exp@one@n\MT@declare@sets
1894             {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
1895         }%
1896       }%
1897     }%
1898   }%
1899 }

\MT@curr@set@name We need to remember the name of the set currently being declared.
1900 \let\MT@curr@set@name\empty

\MT@declare@sets Define the current set name and parse the keys.
1901 \def\MT@declare@sets#1#2#3{%
1902   \KV@sp@def\MT@curr@set@name{#2}%
1903   \MT@ifdefined@nT{MT@#1@set@@\MT@curr@set@name}{%
1904     \MT@warning{Redefining set `~\MT@curr@set@name'}%
1905   }%
1906   \global\MT@let@nc{MT@#1@set@@\MT@curr@set@name}\@empty
1907   debug \MT@info{1}{declaring \nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
1908   \setkeys{MT@#1@set}{#3}%
1909 }

\MT@define@set@keys Define the keyval keys for font sets.
1910 \def\MT@define@set@keys#1{%
1911   \MT@define@set@key@{encoding}{#1}%
1912   \MT@define@set@key@{family}{#1}%
1913   \MT@define@set@key@{series}{#1}%
1914   \MT@define@set@key@{shape}{#1}%
1915   \MT@define@set@key@{size}{#1}%
1916   \MT@define@set@key@{font}{#1}%
1917 }

\MT@define@set@key@ <#1> = font axis, <#2> = feature.
1918 \def\MT@define@set@key@#1#2{%
1919   \define@key{MT@#2@set}{#1}[]{%
1920     \global\MT@let@nc{MT@#2list@#1@\MT@curr@set@name}\empty
1921     \MT@map@clist@n{##1}{%
1922       \KV@sp@def\MT@val{####1}%
1923       \MT@get@highlevel{#1}%

We do not add the expanded value to the list ...
1924       \MT@exp@two@n\g@addto@macro
1925         {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
1926         {\MT@val,}%
1927     }%

... but keep in mind that the list has to be expanded at the end of the preamble.
1928   \edef\x{%
1929     \expandafter\noexpand\expandafter\MT@fix@font@spec

```

```

1930      \expandafter\noexpand\csname MT@#2list@#1@MT@curr@set@name\endcsname
1931    }%
1932    \MT@exp@one@n\MT@addto@setup\x
1933 <debug>\MT@dinfo@n1{1}{-- #1: \@nameuse{MT@#2list@#1@MT@curr@set@name}}%
1934  }%
1935 }

```

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

```

1936 \def\MT@get@highlevel#1{%
1937   \expandafter\MT@test@ast\MT@val*\@nil{%

```

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

```

1938   \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
1939   \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.

```

1940 }%
1941 }

```

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

```

1942 \def\MT@test@ast#1*#2\@nil{%
1943   \def\@tempa{#1}%
1944   \MT@ifempty{#2}%
1945     \@gobble
1946     \@firstofone
1947 }

```

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

```

1948 \def\MT@fix@font@spec#1{%
1949   \xdef#1{#1}%
1950   \global\MT@make@string#1%
1951 }

```

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

```

1952 \def\MT@define@set@key@size#1{%
1953   \define@key{MT@#1@set}{size}[]{%
1954     \MT@map@clist@n{##1}{%
1955       \KV@sp@def\MT@val{###1}%
1956       \expandafter\MT@get@range\MT@val--\@nil
1957       \ifx\MT@val\relax \else
1958         \expandafter\MT@xadd
1959         \csname MT@#1list@size@\MT@curr@set@name\endcsname
1960         {{{\MT@lower}{\MT@upper}\relax}}%
1961       \fi
1962     }%
1963 <debug>\MT@dinfo@n1{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
1964  }%
1965 }

```

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

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



```

1966 \def\MT@get@range#1-#2-#3\@nil{%
1967   \MT@ifempty{#1}{%
1968     \MT@ifempty{#2}{%
1969       \let\MT@val\relax
1970     }{%
1971       \def\MT@lower{0}%
1972       \def\MT@val{#2}%
1973       \MT@get@size
1974       \edef\MT@upper{\MT@val}%
1975     }%
1976   }{%
1977     \def\MT@val{#1}%
1978     \MT@get@size
1979     \ifx\MT@val\relax \else
1980       \edef\MT@lower{\MT@val}%
1981       \MT@ifempty{#2}{%
1982         \MT@ifempty{#3}{%
1983           \def\MT@upper{-1}%
1984         }{%

```

2048pt is  $\text{\TeX}$ 's maximum font size.

```

1985   \def\MT@upper{2048}%
1986   }%
1987 }{%
1988   \def\MT@val{#2}%
1989   \MT@get@size
1990   \ifx\MT@val\relax \else
1991     \MT@ifdim\MT@lower>\MT@val{%
1992       \MT@warning{%
1993         Invalid size range (\MT@lower\space > \MT@val) in font set
1994         ~\MT@curr@set@name'.\MessageBreak Swapping sizes}%
1995       \edef\MT@upper{\MT@lower}%
1996       \edef\MT@lower{\MT@val}%
1997     }{%
1998       \edef\MT@upper{\MT@val}%
1999     }%
2000     \MT@ifdim\MT@lower=\MT@upper{%
2001       \def\MT@upper{-1}%
2002     }\relax
2003   \fi
2004 }%
2005 \fi
2006 }%
2007 }

```

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

```

2008 \def\MT@get@size{%

```

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

```

2009   \if*\MT@val\relax
2010     \def\@tempa{\normalsize}%
2011   \else
2012     \MT@let@cn\@tempa{\MT@val}%
2013   \fi
2014   \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).

```

2015 \begingroup
2016   \def\set@fontsize##1##2##3##4\@nil{\gdef\MT@val{##2}}%
2017   \@tempa\@nil
2018 \endgroup
2019 \fi

```

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

```

2020 \MT@ifdimen\MT@val{%
2021   \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2022   \edef\MT@val{\strip@pt\@tempdima}%
2023 }{%
2024   \MT@warning{Could not parse font size `~\MT@val'\MessageBreak
2025               in font set `~\MT@curr@set@name'}%
2026   \let\MT@val\relax
2027 }%
2028 }

```

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

```

2029 \def\MT@define@set@key@font#1{%
2030   \define@key{MT@#1@set}{font}[]{%
2031     \global\MT@let@nc{MT@#1list@font@\MT@curr@set@name}\@empty
2032     \MT@map@clist@n{##1}{%
2033       \KV@sp@def\MT@val{###1}%
2034       \expandafter\MT@get@font\MT@val////\@nil
2035       \MT@exp@two@n@g@addto@macro
2036         {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
2037         {\MT@val},}%
2038     }%
2039   \edef\x{%
2040     \expandafter\noexpand\expandafter\MT@fix@font@spec
2041     \expandafter\noexpand\csname MT@#1list@font@\MT@curr@set@name\endcsname
2042   }%
2043   \MT@exp@one@n\MT@addto@setup\x
2044   (debug) \MT@info@n{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
2045   }%
2046 }

```

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

```

2047 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2048   \MT@ifempty{#1#2#3#4#5}\relax{%
2049     \let\@tempb\@empty
2050     \def\MT@temp{#1/#2/#3/#4/#5}%
2051     \MT@get@axis{encoding}{#1}%
2052     \MT@get@axis{family}{#2}%
2053     \MT@get@axis{series}{#3}%
2054     \MT@get@axis{shape}{#4}%
2055     \MT@ifempty{#5}{%
2056       \MT@warn@axis@empty{size}{\string\normalsize}%
2057       \def\MT@val{*}%
2058     }{%
2059       \def\MT@val{#5}%
2060     }%
2061     \MT@get@size
2062     \ifx\MT@val\relax\def\MT@val{0}\fi
2063     \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2064     \let\MT@val\@tempb
2065   }%
2066 }

```

`\MT@get@axis`

```

2067 \def\MT@get@axis#1#2{%
2068   \def\MT@val{#2}%
2069   \MT@get@highlevel{#1}%
2070   \MT@ifempty\MT@val{%
2071     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2072     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2073   }\relax
2074   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2075 }

```

`\MT@warn@axis@empty`

```

2076 \def\MT@warn@axis@empty#1#2{%
2077   \MT@warning{#1 axis is empty in font specification\MessageBreak
2078     `~\MT@temp'. Using `~#2' instead}%
2079 }

```

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

```
2080 \MT@map@clist@c\MT@features{\MT@define@set@keys{#1}}
```

It is also used for `\DisableLigatures`.

```
2081 \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.

```

2082 \renewcommand*\UseMicrotypeSet[2][]{%
2083   \KV@sp@def\@tempa{#1}%
2084   \MT@ifempty\@tempa{%
2085     \MT@map@clist@c\MT@features{\MT@use@set{##1}{#2}}}%
2086   }{%
2087     \MT@map@clist@c\@tempa{%
2088       \KV@sp@def\@tempa{##1}%
2089       \MT@ifempty\@tempa\relax{%
2090         \MT@is@feature{activation of set `~#2'}{%
2091           \MT@exp@one@n\MT@use@set
2092             {\csname MT@rbba@\@tempa\endcsname}{#2}}%
2093         }%
2094       }%
2095     }%
2096   }%
2097 }

```

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

```

\MT@ex@setname 2098 \def\MT@use@set#1#2{%
\MT@use@set    2099   \KV@sp@def\@tempa{#2}%
2100   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
2101     \global\MT@edef@n{MT@#1@setname}{\@tempa}%
2102     \MT@info{Using \@nameuse{MT@abbr@#1} set `~\@tempa'}%
2103   }{%
2104     \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2105       \global\MT@edef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2106     }%
2107     \MT@warning{%
2108       The \@nameuse{MT@abbr@#1} set `~\@tempa' is undeclared.\MessageBreak
2109       Using set `~\@nameuse{MT@#1@setname}' instead}%
2110   }%
2111 }

```

`\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.

```

2112 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2113   \KV@sp@def\@tempa{#1}%
2114   \MT@ifempty\@tempa{%
2115     \MT@map@clist@MT@features{{\MT@set@default@set{##1}{#2}}}%
2116   }{%
2117     \MT@map@clist@MT@tempa{{%
2118       \KV@sp@def\@tempa{#1}%
2119       \MT@ifempty\@tempa\relax{%
2120         \MT@is@feature{declaration of default set `#2'}{%
2121           \MT@exp@one@MT@set@default@set
2122             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2123         }%
2124       }%
2125     }}%
2126   }%
2127 }

\MT@default@pr@set
\MT@default@ex@set 2128 \def\MT@set@default@set#1#2{%
\MT@default@kn@set 2129   \KV@sp@def\@tempa{#2}%
2130   \MT@ifdefined@TF{MT@#1@set@@\@tempa}{%
\MT@default@sp@set 2131   debug \MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set \@tempa'}%
\MT@set@default@set 2132   \global\MT@edef@MT@default@#1@set{\@tempa}%
2133   }{%
2134     \MT@warning{%
2135       The \@nameuse{MT@abbr@#1} set \@tempa' is not declared.\MessageBreak
2136       Cannot make it the default set. Using set\MessageBreak `all' instead}%
2137     \global\MT@edef@MT@default@#1@set{all}%
2138   }%
2139 }

```

**\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.

```

2140 \renewcommand*\DeclareMicrotypeAlias[2][]{%
2141   \KV@sp@def\@tempa{#1}%
2142   \KV@sp@def\@tempb{#2}%
2143   \MT@make@string\@tempb
2144   \MT@ifdefined@TF{MT@\@tempa @alias}{%
2145     \MT@warning{Alias font family \@tempb' will override
2146       alias \@nameuse{MT@\@tempa @alias}'\MessageBreak
2147       for font family \@tempa'}%
2148   \global\MT@edef@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.

```

2149   \MT@ifdefined@TF{MT@family}{%
2150   debug \MT@dinfo{1}{Activating alias font \@tempb' for \@tempa'}%
2151   \global\let\MT@familyalias\@tempb
2152   }%
2153 }

```

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

```

2154 \def\LoadMicrotypeFile#1{%
2155   \KV@sp@def\@tempa{#1}%
2156   \MT@make@string\@tempa
2157   \MT@exp@one@MT@in@clist\@tempa\MT@file@list
2158   \ifMT@inlist@
2159     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2160   \else

```

```

2161 \MT@xadd\MT@file@list{\@tempa,}%
2162 \MT@begin@catcodes
2163 \InputIfFileExists{mt-\@tempa.cfg}{%
2164 \edef\MT@curr@file{mt-\@tempa.cfg}%
2165 \MT@vinfo{... Loading configuration file \MT@curr@file}%
2166 }{%
2167 \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2168 does not exist}%
2169 }%
2170 \MT@end@catcodes
2171 \fi
2172 }

```

`\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'.

```

2173 \MT@requires@pdfTeX5{
2174 \renewcommand*\DisableLigatures[1]{%
2175 \edef\MT@active@features{\MT@active@features,nl}%
2176 \MT@noligaturestrue
2177 \MT@declare@sets{nl}{no ligatures}{#1}%
2178 \gdef\MT@nl@setname{no ligatures}%
2179 }
2180 }{

```

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

```

2181 \renewcommand*\DisableLigatures[1]{%
2182 \MT@warning{Disabling ligatures of a font is only possible\MessageBreak
2183 with pdfTeX version 1.30 or newer.\MessageBreak
2184 Ignoring \string\DisableLigatures}%
2185 \global\let\DisableLigatures\@gobble
2186 }
2187 }

```

### 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.

```

2188 <beta>
2189 \def\DeclareMicrotypeBabelHook#1#2{%
2190 \MT@map@clist@n{#1}{%
2191 \KV@esp@def\@tempa{##1}%
2192 \global\MT@def@n{\MT@babel@\@tempa}{#2}%
2193 }%
2194 }
2195 </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 attributes and list of character protrusion factors.

`\MT@pr@c@name` A new macro called `\MT@pr@c@<name>` will be defined to be `<#3>` (i.e. the list of characters, not expanded).

`\MT@extra@load`

`\MT@extra@factor`

`\MT@extra@preset` 2196 \renewcommand\*\SetProtrusion[2][ ]{%

`\MT@extra@unit` 2197 \let\MT@pr@c@name\@undefined

`\MT@extra@context`

`\MT@permutelist`

```

2198 \let\MT@extra@context\@empty
2199 \MT@map@clist@n{load,factor,unit,preset}{\MT@let@nc{MT@extra@##1}\@undefined}%

```

Parse the optional first argument:

```
2200 \setkeys{MT@pr@c}{#1}%
```

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

```
2201 \MT@get@codes@name{pr}%
```

Extra options.

```

2202 \MT@map@clist@n{factor,unit,preset}{\MT@set@opt{pr}{##1}}%
2203 (debug)\MT@info{1}{creating protrusion list ~\MT@pr@c@name'}%
2204 \def\MT@permutelist{pr@c}%
2205 \setkeys{MT@cfg}{#2}%

```

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

```
2206 \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.

```

2207 \MT@begin@catcodes
2208 \MT@set@pr@list
2209 }

```

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

```

2210 \def\MT@set@pr@list#1{%
2211 \global\MT@def@n{MT@pr@c@\MT@pr@c@name}{#1}%
2212 \MT@end@catcodes
2213 }

```

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

```

\MT@extra@load 2214 \renewcommand*\SetExpansion[2][]{%
\MT@extra@factor 2215 \let\MT@ex@c@name\@undefined
\MT@extra@stretch 2216 \let\MT@extra@context\@empty
\MT@extra@shrink 2217 \MT@map@clist@n{load,factor,preset,stretch,shrink,step,auto}%
\MT@extra@step 2218 {\MT@let@nc{MT@extra@##1}\@undefined}%
\MT@extra@auto 2219 \setkeys{MT@ex@c}{#1}%
\MT@extra@preset 2220 \MT@get@codes@name{ex}%
\MT@extra@context

```

The extra options to \SetExpansion also have to be dealt with only after we know the name.

```

2221 \MT@map@clist@n{preset,stretch,shrink,step,auto}{\MT@set@opt{ex}{##1}}%
2222 \MT@ifdefined@cT\MT@extra@factor{%
2223 \ifnum\MT@extra@factor>\m
2224 \MT@warning@n{Expansion factor \number\MT@extra@factor\space too
2225 large in list\MessageBreak ~\MT@ex@c@name'. Setting it to the
2226 maximum of 1000}%
2227 \let\MT@extra@factor\m
2228 \fi
2229 \global\MT@let@nc{MT@ex@c@\MT@ex@c@name @factor}\MT@extra@factor
2230 }%
2231 (debug)\MT@info{1}{creating expansion list ~\MT@ex@c@name'}%
2232 \def\MT@permutelist{ex@c}%
2233 \setkeys{MT@cfg}{#2}%
2234 \MT@permute
2235 \MT@begin@catcodes
2236 \MT@set@ex@list

```

```

2237 }

\MT@set@ex@list Same story.
2238 \def\MT@set@ex@list#1{%
2239   \global\MT@def@n{MT@ex@c@MT@ex@c@name}{#1}%
2240   \MT@end@catcodes
2241 }

2242 <beta>

\SetExtraSpacing
  \MT@sp@c@name 2243 \renewcommand*\SetExtraSpacing[2][{}]{%
  \MT@extra@load 2244   \let\MT@sp@c@name\undefined
  \MT@extra@load 2245   \let\MT@extra@context\empty
  \MT@extra@factor 2246   \MT@map@clist@n{load,factor,unit,preset}{\MT@let@c{MT@extra@##1}\undefined}%
  \MT@extra@unit 2247   \setkeys{MT@sp@c}{#1}%
  \MT@extra@preset 2248   \MT@get@codes@name{sp}%
  \MT@extra@context 2249   \MT@map@clist@n{factor,unit,preset}{\MT@set@opt{sp}{##1}}%
  2250 <debug>\MT@dinfo{1}{creating spacing list `~\MT@sp@c@name'}%
  2251   \def\MT@permutelist{sp}%
  2252   \setkeys{MT@cfg}{#2}%
  2253   \MT@permute
  2254   \MT@begin@catcodes
  2255   \MT@set@sp@list
  2256 }

\MT@set@sp@list
2257 \def\MT@set@sp@list#1{%
2258   \global\MT@def@n{MT@sp@c@MT@sp@c@name}{#1}%
2259   \MT@end@catcodes
2260 }

\SetExtraKerning
  \MT@kn@c@name 2261 \renewcommand*\SetExtraKerning[2][{}]{%
  \MT@extra@load 2262   \let\MT@kn@c@name\undefined
  \MT@extra@load 2263   \let\MT@extra@context\empty
  \MT@extra@factor 2264   \MT@map@clist@n{load,factor,unit,preset}{\MT@let@c{MT@extra@##1}\undefined}%
  \MT@extra@unit 2265   \setkeys{MT@kn@c}{#1}%
  \MT@extra@preset 2266   \MT@get@codes@name{kn}%
  \MT@extra@context 2267   \MT@map@clist@n{factor,unit,preset}{\MT@set@opt{kn}{##1}}%
  2268 <debug>\MT@dinfo{1}{creating kerning list `~\MT@kn@c@name'}%
  2269   \def\MT@permutelist{kn}%
  2270   \setkeys{MT@cfg}{#2}%
  2271   \MT@permute
  2272   \MT@begin@catcodes
  2273   \MT@set@kn@list
  2274 }

\MT@set@kn@list
2275 \def\MT@set@kn@list#1{%
2276   \global\MT@def@n{MT@kn@c@MT@kn@c@name}{#1}%
2277   \MT@end@catcodes
2278 }

2279 </beta>

\MT@get@codes@name Use file name and line number as the list name if the user didn't bother creating
one.
2280 \def\MT@get@codes@name#1{%
2281   \MT@ifdefined@n@TF{MT@#1@c@name}{%
2282     \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%

```

```

2283 \MT@warning{Redefining list ~\@nameuse{MT@#1@c@name}}}%
2284 }%
2285 }{%
2286 \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2287 }%
2288 \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.

```

2289 \MT@ifdefined@c@T\MT@extra@load{%
2290 \global\MT@let@nc{MT@#1@c@\MT@curr@set@name load}\MT@extra@load
2291 }%
2292 }

```

`\MT@set@opt` The additional options can also only be dealt with after we know the list name.

```

2293 \def\MT@set@opt#1#2{%
2294 \MT@ifdefined@n@T{MT@extra@#2}{%
2295 \global\MT@let@nn{MT@#1@c@\csname MT@#1@c@name\endcsname @#2}{MT@extra@#2}%
2296 }%
2297 }

```

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

```

2298 \def\MT@define@code@key#1#2{%
2299 \define@key{MT@#2}{#1}[]{%
2300 \@tempcnta=\@ne
2301 \MT@map@clist@n{##1}{%
2302 \KV@@sp@def\MT@val{####1}%

```

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

```

2303 \MT@get@highlevel{#1}%
2304 \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
2305 \advance\@tempcnta \@ne
2306 }%
2307 }%
2308 }

```

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

```

2309 \def\MT@define@code@key@size#1{%
2310 \define@key{MT@#1}{size}[]{%
2311 \MT@map@clist@n{##1}{%
2312 \KV@@sp@def\MT@val{####1}%
2313 \expandafter\MT@get@range\MT@val--\@nil
2314 \ifx\MT@val\relax \else
2315 \expandafter\MT@xadd\csname MT@tempsize\endcsname
2316 {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2317 \fi
2318 }%
2319 }%
2320 }

```

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

```

2321 \def\MT@define@code@key@font#1{%
2322 \define@key{MT@#1}{font}[]{%
2323 \MT@map@clist@n{##1}{%
2324 \KV@@sp@def\MT@val{####1}%
2325 \expandafter\MT@get@font@and@size\MT@val////\@nil
2326 \global\MT@edef@n{MT@MT@permutelist @\@tempb\MT@extra@context}%
2327 {\csname MT@MT@permutelist @name\endcsname}%
2328 (*debug)
2329 \MT@dinfo@n{1}{initializing: use list for font \@tempb=\MT@val

```



```

2330             \ifx\MT@extra@context\@empty\else\MessageBreak
2331             (context: \MT@extra@context)\fi}%
2332 
```

*/debug*

```

2333     \expandafter\MT@xaddb
2334     \csname MT@MT@permutelist @\@tempb\MT@extra@context @sizes\endcsname
2335     {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
2336     }%
2337 }%
2338 }

```

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

```

2339 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2340   \MT@ifempty{#1#2#3#4#5}\relax{%
2341     \let\@tempb\@empty
2342     \def\MT@temp{#1/#2/#3/#4/#5}%
2343     \MT@get@axis{encoding}{#1}%
2344     \MT@get@axis{family}{#2}%
2345     \MT@get@axis{series}{#3}%
2346     \MT@get@axis{shape}{#4}%

```

Append an asterisk for the size.

```

2347   \edef\@tempb{\@tempb*}%
2348   \MT@ifempty{#5}%
2349   \MT@warn@axis@empty{size}{\string\normalsize}%
2350   \def\MT@val{*}%
2351   }{%
2352   \def\MT@val{#5}%
2353   }%
2354   \MT@get@size
2355   }%
2356 }

2357 \MT@define@code@key{encoding}{cfg}
2358 \MT@define@code@key{family}{cfg}
2359 \MT@define@code@key{series}{cfg}
2360 \MT@define@code@key{shape}{cfg}
2361 \MT@define@code@key@size{cfg}
2362 \MT@define@code@key@font{cfg}

```

`\MT@declare@codes` The options in the optional first argument. We wouldn't need to define these keys for each feature, if we were using the `xkeyval` package.

```

2363 \def\MT@declare@codes#1{%
2364   \define@key{MT@#1@c}{name}[]{}%
2365   \MT@ifempty{#1}\relax{\MT@defn{MT@#1@c@name}{##1}}%
2366   \define@key{MT@#1@c}{load}[]{}%
2367   \MT@ifempty{#1}\relax{\def\MT@extra@load{##1}}%
2368   \define@key{MT@#1@c}{factor}[]{}%
2369   \MT@ifempty{#1}\relax{\def\MT@extra@factor{##1}}%
2370   \define@key{MT@#1@c}{preset}[]{}%
2371   \MT@ifempty{#1}\relax{\def\MT@extra@preset{##1}}%

```

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

```

2372   \define@key{MT@#1@c}{context}[]{}%
2373   \MT@ifempty{#1}\relax{\def\MT@extra@context{##1}}%
2374 }

```

```

2375 \MT@map@clist@c\MT@features{\MT@declare@codes{#1}}

```

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

```

2376 \define@key{MT@pr@c}{unit}[character]{}%
2377 \let\MT@extra@unit\@empty
2378 \KV@sp@def\@tempa{#1}%

```

```

2379 \MT@ifstreq\@tempa{relative}{%
2380 \MT@warning{Value `relative' for key `unit' is deprecated.\MessageBreak
2381 Use `unit=character' instead. For now, I'll do it.\MessageBreak
2382 for you}%
2383 \def\@tempa{character}%
2384 }\relax
2385 \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.

```

2386 \MT@ifdimen\@tempa{%
2387 \let\MT@extra@unit\@tempa
2388 }{%
2389 \MT@warning{`\@tempa' is not a dimension.\MessageBreak
2390 Ignoring it and setting values relative to.\MessageBreak
2391 character widths}%
2392 }%
2393 }%
2394 }

```

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

```

2395 <beta>
2396 \def\MT@define@key@unit#1{%
2397 \define@key{MT@#1@c}{unit}[space]{%
2398 \let\MT@extra@unit\empty
2399 \KV@sp@def\@tempa{##1}%
2400 \MT@ifstreq\@tempa{character}\relax{%
2401 \let\MT@extra@unit\mone
2402 \MT@ifstreq\@tempa{space}\relax{%
2403 \MT@ifdimen\@tempa{%
2404 \let\MT@extra@unit\@tempa
2405 }{%
2406 \MT@warning{`\@tempa' is not a dimension.\MessageBreak
2407 Ignoring it and setting values relative to.\MessageBreak
2408 width of space}%
2409 }%
2410 }%
2411 }%
2412 }%
2413 }
2414 \MT@define@key@unit{sp}
2415 \MT@define@key@unit{kn}
2416 </beta>

```

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

```

2417 \def\MT@define@ex@key#1{%
2418 \define@key{MT@ex@c}{#1}[]{%
2419 \MT@ifempty{##1}\relax{%
2420 \MT@ifint{##1}{%

```

A space terminates the number.

```

2421 \MT@def@n{MT@extra@#1}{##1 }%
2422 }{%
2423 \MT@warning{%
2424 Value `##1' for option `#1' is not a number.\MessageBreak
2425 Ignoring it}%
2426 }%
2427 }%
2428 }%
2429 }

```

```

2430 \MT@define@ex@key{stretch}
2431 \MT@define@ex@key{shrink}
2432 \MT@define@ex@key{step}
2433 \define@key{MT@ex@}{auto}[true]{%
2434   \KV@sp@def\@tempa{#1}%
2435   \csname if\@tempa\endcsname

```

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

```

2436   \MT@requires@pdftex4{%
2437     \def\MT@extra@auto{autoexpand}%
2438   }{%
2439     \MT@warning{pdfTeX too old for automatic font expansion}%
2440   }
2441   \else
2442     \MT@requires@pdftex4{%
2443       \let\MT@extra@auto\@empty
2444     }\relax
2445   \fi
2446 }

```

### 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.

```

2447 \renewcommand*\DeclareCharacterInheritance[1][]{%
2448   \let\MT@extra@context\@empty
2449   \KV@sp@def\@tempa{#1}%
2450   \MT@begin@catcodes
2451   \MT@set@inh@list
2452 }

```

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

```

2453 \def\MT@set@inh@list#1#2{%
2454   \MT@ifempty\@tempa{%
2455     \MT@map@clist@c\MT@features{\MT@declare@char@inh{##1}{#1}{#2}}}%
2456   }{%
2457     \MT@map@clist@c\@tempa{%
2458       \KV@sp@def\@tempa{#1}%
2459       \MT@ifempty\@tempa\relax{%
2460         \MT@is@feature{inheritance declaration for `#1'}{%
2461           \MT@exp@one@n\MT@declare@char@inh
2462             {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
2463         }%
2464       }%
2465     }%
2466   }%
2467   \MT@end@catcodes
2468 }

```

`\MT@declare@char@inh` The optional argument may be used to restrict the inheritance list to a feature. The lists cannot be given a name by the user.

```

2469 \def\MT@declare@char@inh#1#2#3{%
2470   \MT@edefn{MT@#1@inh@name}%
2471     {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%

```

```

2472 \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
2473 (debug)\MT@info{1}{creating inheritance list `\'@nameuse{MT@#1@inh@name}'}%
2474 \global\MT@def@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
2475 \def\MT@permute@list{#1@inh}%
2476 \setkeys{MT@inh}{#2}%
2477 \MT@permute
2478 }

```

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

```

2479 \define@key{MT@inh}{encoding}[]{%
2480 \def\MT@val{#1}%
2481 \expandafter\MT@encoding@check\MT@val,\@nil
2482 \MT@get@highlevel{encoding}%
2483 \MT@edef@n{MT@tempencoding1}{\MT@val}%
2484 }

```

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

```

2485 \def\MT@encoding@check#1,#2\@nil{%
2486 \MT@ifempty{#2}\relax{%
2487 \edef\MT@val{#1}%
2488 \MT@warning{You may only specify one encoding for character\MessageBreak
2489 inheritance lists. Ignoring encoding(s) #2}%
2490 }%
2491 }

```

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

```

2492 \MT@define@code@key{family}{inh}
2493 \MT@define@code@key{series}{inh}
2494 \MT@define@code@key{shape}{inh}
2495 \MT@define@code@key@size{inh}
2496 \MT@define@code@key@font{inh}

```

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

```

2497 \def\MT@inh@do#1,{%
2498 \ifx\relax#1\@empty \else
2499 \MT@inh@split #1==\relax
2500 \expandafter\MT@inh@do
2501 \fi
2502 }

```

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

```

2503 \def\MT@inh@split#1=#2=#3\relax{%
2504 \def\@tempa{#1}%
2505 \ifx\@tempa\@empty \else
2506 \MT@get@slot
2507 \ifnum\MT@char > \m@ne
2508 \let\MT@val\MT@char
2509 \MT@map@clist@n{#2}{%
2510 \def\@tempa{##1}%
2511 \ifx\@tempa\@empty \else
2512 \MT@get@slot
2513 \ifnum\MT@char > \m@ne
2514 \expandafter\MT@exadd

```

```

2515         \csname MT@inh@MT@inh@name @\MT@val @\endcsname
2516         {{\MT@char}}}%
2517     \fi
2518 \fi
2519 }%
2520 <debug>
2521     \MT@info@n1{2}{children of #1 (\MT@val):
2522         \@nameuse{MT@inh@MT@inh@name @\MT@val @}}%
2523 </debug>
2524 \fi
2525 \fi
2526 }

```

### 13.5.5 Permutation

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

```

2527 \def\MT@permute{%
2528     \let\MT@cnt@encoding\@ne
2529     \MT@permute@

```

Undefine commands for the next round.

```

2530     \MT@permute@reset
2531 }
2532 \def\MT@permute@{%
2533     \let\MT@cnt@family\@ne
2534     \MT@permute@@
2535     \MT@increment\MT@cnt@encoding
2536     \MT@ifdefined@nT{MT@tempencoding\MT@cnt@encoding}%
2537     \MT@permute@
2538 }
2539 \def\MT@permute@@{%
2540     \let\MT@cnt@series\@ne
2541     \MT@permute@@@
2542     \MT@increment\MT@cnt@family
2543     \MT@ifdefined@nT{MT@tempfamily\MT@cnt@family}%
2544     \MT@permute@@@
2545 }
2546 \def\MT@permute@@@{%
2547     \let\MT@cnt@shape\@ne
2548     \MT@permute@@@@
2549     \MT@increment\MT@cnt@series
2550     \MT@ifdefined@nT{MT@tempseries\MT@cnt@series}%
2551     \MT@permute@@@@
2552 }
2553 \def\MT@permute@@@@{%
2554     \MT@permute@@@@@
2555     \MT@increment\MT@cnt@shape
2556     \MT@ifdefined@nT{MT@tempshape\MT@cnt@shape}%
2557     \MT@permute@@@@@
2558 }

```

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

```

2559 \def\MT@permute@@@@@{%

```

```

2560 \MT@permute@define{encoding}%
2561 \ifMT@document
2562   \ifx\MT@tempencoding\empty \else
2563     \MT@ifdefined@n@TF{T@MT@tempencoding}\relax
2564     {\expandafter\expandafter\expandafter\@gobble}%
2565   \fi
2566 \fi
2567 \MT@permute@@@@@
2568 }

\MT@permute@@@@@

```

```

2569 \def\MT@permute@@@@@{%
2570   \MT@permute@define{family}%
2571   \MT@permute@define{series}%
2572   \MT@permute@define{shape}%
2573   \edef\@tempa{\MT@tempencoding
2574               /\MT@tempfamily
2575               /\MT@tempseries
2576               /\MT@tempshape
2577               /\MT@ifdefined@c@T\MT@tempsize *}%

```

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

```

2578 \def\@tempb{////}%
2579 \ifx\@tempa\@tempb \else
2580   \ifx\MT@tempencoding\empty
2581     \MT@warning{%
2582       You have to specify an encoding for\MessageBreak
2583       \@nameuse{MT@abbr@MT@permutelist} list
2584       ~\@nameuse{MT@MT@permutelist @name}'.\MessageBreak
2585       Ignoring it}%
2586   \else
2587     \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

2588   \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
2589     \MT@map@tlist@c
2590     \MT@tempsize
2591     \MT@check@rlist
2592   }%
2593   \expandafter\MT@xaddb
2594   \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
2595   \MT@tempsize
2596 \<debug>
2597   \MT@info@n1{1}{initializing: use list for font \@tempa,\MessageBreak
2598     sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
2599       @sizes\endcsname}%
2600 \</debug>
2601   }{%

```

Only one list should apply to a given combination.

```

2602   \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
2603     \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
2604       ~\@nameuse{MT@MT@permutelist @name}' will override list\MessageBreak
2605       ~\@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
2606       for font \@tempa}%
2607   }%
2608 \<debug>
2609   \MT@info@n1{1}{initializing: use list for font \@tempa
2610     \ifx\MT@extra@context\empty\else\MessageBreak
2611     (context: \MT@extra@context)\fi}%

```

```

2612 /debug
2613 }%
2614 \global\MT@edef\MT@MT@permutelist @\@tempa\MT@extra@context}%
2615 {\csname MT@MT@permutelist @name\endcsname}%
2616 \fi
2617 \fi
2618 }

```

\MT@permute@define Define the commands.

```

2619 \def\MT@permute@define#1{%
2620 \expandafter\@tempcnta=\csname MT@cnt@#1\endcsname\relax
2621 \MT@ifdefined\MT@temp#1\the\@tempcnta}%
2622 {\MT@edef\MT@temp#1{\csname MT@temp#1\the\@tempcnta\endcsname}}%
2623 {\MT@let\MT@temp#1\@empty}%
2624 }

```

\MT@permute@reset Reset the commands.

```

2625 \def\MT@permute@reset{%
2626 \MT@permute@reset@{encoding}%
2627 \MT@permute@reset@{family}%
2628 \MT@permute@reset@{series}%
2629 \MT@permute@reset@{shape}%
2630 \let\MT@tempsize\@undefined
2631 }

```

\MT@permute@reset@

```

2632 \def\MT@permute@reset@#1{%
2633 \@tempcnta=\@ne
2634 \MT@loop
2635 \MT@let\MT@temp#1\the\@tempcnta\@undefined
2636 \advance\@tempcnta\@ne
2637 \MT@ifdefined\MT@temp#1\the\@tempcnta}%
2638 \iftrue
2639 \iffalse
2640 \MT@repeat
2641 }

```

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

```

2642 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

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

```

2643 \def\MT@check@rlist@#1#2#3{%
2644 \def\@tempb{#1}%
2645 \def\@tempc{#2}%
2646 \@tempswafalse
2647 \expandafter\MT@map@tlist@
2648 \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
2649 \MT@check@range
2650 }

```

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

```

2651 \def\MT@check@range#1{\expandafter\MT@check@range@ #1}

```

\MT@check@range@ \@tempb and \@tempc are lower resp. upper bound of the new range, <#2> and <#3> those of the existing range.

```

2652 \def\MT@check@range@#1#2#3{%
2653 \MT@ifdim{#2}=\m@ne%
2654 \MT@ifdim\@tempc=\m@ne%

```

- Both items are simple sizes.

```
2655      \MT@ifdim\@tempb={#1}\@tempswattrue\relax
2656      }{%
```

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

```
2657      \MT@ifdim\@tempb>{#1}\relax{%
2658      \MT@ifdim\@tempc>{#1}{%
2659      \@tempswattrue
2660      \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
2661      }\relax
2662      }%
2663      }%
2664      }{%
2665      \MT@ifdim\@tempc=\m@ne{%
```

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

```
2666      \MT@ifdim\@tempb<{#2}{%
2667      \MT@ifdim\@tempb<{#1}\relax\@tempswattrue
2668      }\relax
2669      }{%
```

- Both items are ranges.

```
2670      \MT@ifdim\@tempb<{#2}{%
2671      \MT@ifdim\@tempc>{#1}{%
2672      \@tempswattrue
2673      \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
2674      }\relax
2675      }\relax
2676      }%
2677      }%
2678      \if@tempswa
2679      \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
2680      ~\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
2681      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.

```
2682      \expandafter\MT@tlist@break
2683      \fi
2684      }
```

## 13.6 Package Options

### 13.6.1 Declaring the Options

`\ifMT@opt@expansion` Keep track of whether the user explicitly set these options.

```
\ifMT@opt@auto 2685 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 2686 \newif\ifMT@opt@auto
2687 \newif\ifMT@opt@DVI
```

`\MT@define@option` expansion and protrusion may be true, false, compatibility, nocompatibility and/or a *<set name>*.

```
2688 \def\MT@define@option#1{%
2689   \define@key{MT}{#1}[true]{%
2690     \csname MT@opt@#1true\endcsname
2691     \MT@map@clist@n{##1}{%
```



```

2692 \KV@@sp@def\MT@val{###1}%
2693 \MT@ifempty\MT@val\relax{%
2694   \csname MT@#1true\endcsname
2695   \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2696   \MT@ifstreq\MT@val{true}\relax
2697   {%
2698     \MT@ifstreq\MT@val{false}{%
2699       \csname MT@#1false\endcsname
2700     }{%
2701       \MT@ifstreq\MT@val{compatibility}{%
2702         \MT@let@nc{MT@\@tempb @level}\@ne
2703       }{%
2704         \MT@ifstreq\MT@val{nocompatibility}{%
2705           \MT@let@nc{MT@\@tempb @level}\tw@
2706         }{%

```

If everything failed, it should be a set name.

```

2707   \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
2708     \global\MT@edef@n{MT@\@tempb @setname}{\MT@val}%
2709   }{%
2710     \global\MT@edef@n{MT@\@tempb @setname}%
2711     {\@nameuse{MT@default@\@tempb @set}}%
2712     \MT@warning@n{%
2713       The #1 set `~\MT@val' is undeclared.\MessageBreak
2714       Using set `~\@nameuse{MT@\@tempb @setname}' instead}%
2715     }%
2716   }%
2717 }%
2718 }%
2719 }%
2720 }%
2721 }%
2722 }%
2723 }
2724 \MT@define@option{protrusion}
2725 \MT@define@option{expansion}

```

activate is a shortcut for protrusion and expansion.

```

2726 \define@key{MT}{activate}[]{%
2727   \setkeys{MT}{protrusion={#1}}%
2728   \setkeys{MT}{expansion={#1}}%
2729 }
2730 {*beta}

```

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

```

2731 \def\MT@define@option@#1{%
2732   \define@key{MT}{#1}[true]{%
2733     \csname MT@opt@#1true\endcsname
2734     \MT@map@clist@n{##1}{%
2735       \KV@@sp@def\MT@val{###1}%
2736       \MT@ifempty\MT@val\relax{%
2737         \csname MT@#1true\endcsname
2738         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2739         \MT@ifstreq\MT@val{true}\relax
2740       }{%
2741         \MT@ifstreq\MT@val{false}{%
2742           \csname MT@#1false\endcsname
2743         }{%
2744           \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
2745             \global\MT@edef@n{MT@\@tempb @setname}{\MT@val}%
2746           }{%

```

```

2747         \global\MT@edef\MT@{\@tempb @setname}%
2748         {\@nameuse\MT@default\@tempb @set}}%
2749         \MT@warning\MT@{\@tempb @set}%
2750         The #1 set '\MT@val' is undeclared.\MessageBreak
2751         Using set '\@nameuse\MT@{\@tempb @setname}' instead}%
2752     }%
2753 }%
2754 }%
2755 }%
2756 }%
2757 }%
2758 }

2759 \MT@define@option@{spacing}
2760 \MT@define@option@{kerning}
2761 \beta

```

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

```

2762 \def\MT@def@bool@opt#1#2{%
2763   \define@key{MT}{#1}[]{}%
2764   \MT@ifempty{#1}%
2765     {\def\@tempa{true}}%
2766     {\def\@tempa{#1}}%
2767   \MT@ifstreq\@tempa{true}\relax{%
2768     \MT@ifstreq\@tempa{false}\relax{%
2769       \MT@warning\MT@{\@tempa}%
2770       `##1' is not an admissible value for option\MessageBreak
2771       `#1'. Assuming `false'}%
2772     \def\@tempa{false}%
2773   }%
2774 }%
2775 #2%
2776 }%
2777 }

```

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

```

2778 \def\MT@def@simple@bool@opt#1{\MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
2779 \MT@map@tlistn{{draft}{auto}{selected}}
2780 \beta {babel}%
2781 } \MT@def@simple@bool@opt

```

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

```

2782 \MT@def@bool@opt{DVIoutput}{%
2783   \csname if\@tempa\endcsname
2784   \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
2785   \pdfoutput\z@
2786   \else
2787   \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
2788   \pdfoutput\@ne
2789   \fi
2790 }

```

Setting the `defersetup` option to false will restore the old behaviour, where the setup took place at the time when the package was loaded. This is undocumented, since I would like to learn about the cases where this is necessary.

The only problem with the new deferred setup I can think of is when a box is being constructed inside the preamble and this box contains a font that is not

loaded before the box is being used.

```

2791 \MT@def@bool@opt{deferssetup}{%
2792   \csname if\@tempa\endcsname \else
2793     \AtEndOfPackage{%
2794       \MT@setup@
2795       \let\MT@setup@\@empty
2796       \let\MT@addto@setup\@firstofone
2797     }%
2798   \fi
2799 }
```

`final` is the opposite to `draft`.

```

2800 \MT@def@bool@opt{final}{%
2801   \csname if\@tempa\endcsname
2802     \MT@draftfalse
2803   \else
2804     \MT@drafttrue
2805   \fi
2806 }
```

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

```

2807 \define@key{MT}{verbose}[]{%
2808   \let\MT@vinfo\MT@info@nl
2809   \MT@ifempty{#1}%
2810     {\def\@tempa{true}}%
2811     {\def\@tempa{#1}}%
2812   \MT@ifstreq\@tempa{true}\relax%
```

Take problems seriously.

```

2813   \MT@ifstreq\@tempa{errors}{%
2814     \let\MT@warning\MT@warn@err
2815     \let\MT@warning@nl\MT@warn@err
2816   }{%
2817     \let\MT@vinfo\@gobble
2818     \MT@ifstreq\@tempa{false}\relax%
2819     \MT@warning@nl{%
2820       `#1' is not an admissible value for option\MessageBreak
2821       `verbose'. Assuming `false'%
2822     }%
2823   }%
2824 }%
2825 }
```

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

```

2826 \def\MT@def@num@opt#1{%
2827   \define@key{MT}{#1}[]{%
2828     \MT@ifempty{#1}%
2829       {\MT@let@cn\@tempa{MT@#1@default}}%
2830     {\def\@tempa{#1}}%
2831   }
```

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

```

2831   \MT@ifint\@tempa{%
2832     \MT@edef@n{MT@#1}{\@tempa}%
2833   }{\MT@warning@nl{%
2834     Value `##1' for option `#1' is not a number.\MessageBreak
2835     Using default value of \number\@nameuse{MT@#1@default}}%
2836   }%
2837 }%
2838 }

2839 \MT@map@tlist@n{{stretch}{shrink}{step}}%
```

```
2840 <beta> {letterspacing}%
2841 } \MT@def@num@opt
```

factor will define the protrusion factor only.

```
2842 \define@key{MT}{factor}[]{}%
2843 \MT@ifempty{#1}%
2844 {\let\@tempa\MT@factor@default}%
2845 {\def\@tempa{#1}}%
2846 \MT@ifint\@tempa{%
2847 \edef\MT@pr@factor{\@tempa}%
2848 }{\MT@warning@n1{%
2849 Value `#1' for option `factor' is not a number.\MessageBreak
2850 Using default value of \number\MT@factor@default}%
2851 }%
2852 }
```

Unit for codes.

```
2853 \define@key{MT}{unit}[]{}%
2854 \MT@ifempty{#1}%
2855 {\def\@tempa{character}}%
2856 {\KV@sp@def\@tempa{#1}}%
2857 \MT@ifstreq\@tempa{relative}{%
2858 \MT@warning{Value `relative' for option `unit' is deprecated.\MessageBreak
2859 Use `unit=character' instead. For now, I'll do it.\MessageBreak
2860 for you}%
2861 \def\@tempa{character}%
2862 } \relax
2863 \MT@ifstreq\@tempa{character} \relax{%
2864 \MT@ifdimen\@tempa{%
2865 \let\MT@pr@unit\@tempa
2866 }{%
2867 \MT@warning@n1{\@tempa' is not a dimension. Ignoring it and.\MessageBreak
2868 setting values relative to character widths}%
2869 }%
2870 }%
2871 }
```

### 13.6.2 Reading the Configuration File

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as T<sub>E</sub>X systems are switching to the pdfT<sub>E</sub>X engine even for DVI output, so that the user might not even be aware of the fact that she's running pdfT<sub>E</sub>X.)

```
2872 \MT@protrusiontrue
2873 \ifnum\pdfoutput<\@ne \else
```

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

```
2874 \MT@requires@pdftex4{
2875 \MT@expansiontrue
2876 \MT@autottrue
2877 } \relax
2878 \fi
```

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

However, the config option must of course be evaluated beforehand. We also have

```
\MT@config@file
\MT@get@config
```

to define a no-op for the regular option processing later.

```
2879 \define@key{MT}{config}[]{\relax}
2880 \def\MT@get@config#1config=#2,#3\@nil{%
2881   \MT@ifempty{#2}%
2882   {\def\MT@config@file{\MT@MT.cfg}}%
2883   {\KV@sp@def\MT@config@file{#2.cfg}}%
2884 }
2885 \expandafter\expandafter\expandafter\MT@get@config
2886 \curname opt\@currname.\@currentx\endcsname,config=\@nil
```

Load the file.

```
2887 \IfFileExists{\MT@config@file}{%
2888   \MT@info@nl{Loading configuration file \MT@config@file}%
2889   \MT@begin@catcodes
2890   \let\MT@begin@catcodes\relax
2891   \let\MT@end@catcodes\relax
2892   \let\MT@curr@file\MT@config@file
2893   \input{\MT@config@file}%
2894   \endgroup
2895 }{\MT@warning@nl{%
2896   Could not find configuration file \MT@config@file!\MessageBreak
2897   This will almost certainly cause undesired results.\MessageBreak
2898   Please fix your installation}%
2899 }
```

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

```
2900 \MT@map@clic\c\MT@features{%
2901   \MT@ifdefined@n@TF{MT@default@#1@set}\relax
2902   {\global\MT@def@n{MT@default@#1@set}{all}}%
2903 }
```

### 13.6.3 Hook for Other Packages

`\Microtype@Hook` This hook may be used by font package authors, e. g. to declare alias fonts. If it is defined, it will be executed here, i. e., after the main configuration file has been loaded, and before the package options are evaluated.

This hook was needed in versions prior to 1.9a to overcome the situation that (1) the microtype package should be loaded after all font defaults have been set up (hence, using `\ifpackageloaded` in the font package 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 deferred setup, one could live without this command, however, it remains here since it’s simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

Package authors should check whether the command is already defined so that existing definitions by other packages aren’t overwritten. Example:

```
\def\MinionPro@MT@Hook{\DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}}
\ifpackageloaded{microtype}
  \MinionPro@MT@Hook
  {\@ifundefined{Microtype@Hook}
    {\let\Microtype@Hook\MinionPro@MT@Hook}
    {\g@addto@macro\Microtype@Hook{\MinionPro@MT@Hook}}}
```

`\MicroType@Hook` with a capital T (which only existed in version 1.7) is provided for compatibility reasons. At some point in the future, it will no longer be available, hence it should not be used.

```
2904 \MT@ifdefined@c@T\MicroType@Hook{%
2905   \MT@warning{%
2906     Command \string\MicroType@Hook\space is deprecated.\MessageBreak
2907     Use \string\Microtype@Hook\space instead}\MicroType@Hook}
2908 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook
```

### 13.6.4 Changing Options Later

`\microtypesetup` Inside the preamble, `\microtypesetup` accepts the same options as the package (unless `defersetup=false`). In the document body, it accepts the options: `protrusion`, `expansion` and `activate`, and `spacing` and `kerning`. Specifying font sets is not allowed.

```
2909 \def\microtypesetup{\setkeys{MT}}
2910 \MT@addto@setup{\def\microtypesetup{\setkeys{MTX}}}
2911 \def\MT@define@optionX#1#2{%
2912   \define@key{MTX}{#1}[true]{%
2913     \KV@sp@def\@tempb{#1}%
2914     \MT@map@clist@n{##1}{%
2915       \KV@sp@def\MT@val{###1}%
2916       \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
2917       \MT@ifempty\MT@val\relax{%
2918         \@tempcnta=\m@ne
2919       \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.

```
2920       \MT@checksetup\@tempb{%
2921         \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
2922         \MT@info{Enabling #1
2923           (level \number\csname MT@\@tempb @level\endcsname)}%
2924       }%
2925     }{%
2926       \MT@ifstreq\MT@val{false}{%
2927         \@tempcnta=\z@
2928         \MT@info{Disabling #1}%
2929       }{%
2930         \MT@ifstreq\MT@val{compatibility}{%
2931           \MT@checksetup\@tempb{%
2932             \@tempcnta=\@ne
2933             \MT@let@nc{MT@\@tempb @level}\@ne
2934             \MT@info{Setting #1 to level 1}%
2935           }%
2936         }{%
2937           \MT@ifstreq\MT@val{nocompatibility}{%
2938             \MT@checksetup\@tempb{%
2939               \@tempcnta=\tw@
2940               \MT@let@nc{MT@\@tempb @level}\tw@
2941               \MT@info{Setting #1 to level 2}%
2942             }%
2943           }{%
2944             \MT@warning{%
2945               Value ``\MT@val' for key `#1' not recognized.\MessageBreak
2946               Use any of `true', `false', `compatibility' or\MessageBreak

```

```

2947             `nocompatibility'%
2948         }%
2949     }%
2950 }%
2951 }%
2952 \ifnum\@tempcnta>\m@ne
2953 #2\@tempcnta\relax
2954 \fi
2955 }%
2956 }%
2957 }%
2958 }

```

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

```

2959 \def\MT@checksetup#1{%
2960   \expandafter\csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
2961   \expandafter\@firstofone
2962   \else
2963     \MT@warning{%
2964       You cannot enable \@nameuse{MT@abbr@#1} if it was disabled\MessageBreak
2965       in the package options,}%
2966     \expandafter\@gobble
2967   \fi
2968 }

2969 \MT@define@optionX{protrusion}\pdfprotrudechars
2970 \MT@define@optionX{expansion}\pdfadjustspacing
2971 (*beta)

```

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

```

2972 \def\MT@define@optionX@#1#2{%
2973   \define@key{MTX}{#1}[true]{%
2974     \KV@sp@def\@tempb{#1}%
2975     \MT@map@clist@n{##1}{%
2976       \KV@sp@def\MT@val{###1}%
2977       \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
2978       \MT@ifempty\MT@val\relax{%
2979         \@tempcnta=\m@ne
2980         \MT@ifstreq\MT@val{true}{%
2981           \MT@checksetup\@tempb{%
2982             \@tempcnta=\@ne
2983             \MT@info{Enabling #1}%
2984           }%
2985         }%
2986         \MT@ifstreq\MT@val{false}{%
2987           \@tempcnta=\z@
2988           \MT@info{Disabling #1}%
2989         }%
2990         \MT@warning{%
2991           Value `~\MT@val' for key `~#1' not recognized.\MessageBreak
2992           Use either `true' or `false'%
2993         }%
2994       }%
2995       \ifnum\@tempcnta>\m@ne
2996         #2\relax
2997       \fi
2998     }%
2999   }%
3000 }%
3001 }

3002 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}

```

```

3003 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3004 \pdfappendkern \@tempcnta}
3005 
```

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

```

3010 \def\MT@gobblethree#1#2#3{}
3011 \let\MT@saved@setupfont\MT@setupfont
3012 \define@key{MTX}{disable}[]{}
3013 \MT@info{Inactivate `\' package}%
3014 \let\MT@setupfont\MT@gobblethree
3015 }
3016 \define@key{MTX}{enable}[]{}
3017 \MT@info{Reactivate `\' package}%
3018 \let\MT@setupfont\MT@saved@setupfont
3019 }

```

### 13.6.5 Processing the Options

\MT@ProcessOptionsWithKV Parse options.

```

3020 \def\MT@ProcessOptionsWithKV#1{%
3021 \let\@tempc\relax
3022 \let\KV@tempa\@empty
3023 \MT@map@clist@{\@classoptionslist}%
3024 \def\CurrentOption{##1}%
3025 \MT@ifdefined@n@T{KV@#1\CurrentOption}{%
3026 \edef\KV@tempa{\KV@tempa,\CurrentOption,}%
3027 \expandtwoargs\@removeelement\CurrentOption
3028 \@unusedoptionlist\@unusedoptionlist
3029 }%
3030 }%
3031 \edef\KV@tempa{%
3032 \noexpand\setkeys{#1}{%
3033 \KV@tempa\@optionlist{\@currname.\@current}%
3034 }%
3035 }%
3036 \KV@tempa
3037 \AtEndOfPackage{\let\@unprocessedoptions\relax}%
3038 \let\CurrentOption\@empty
3039 }
3040 \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).

```

3041 \MT@addto@setup{%
3042 \ifMT@draft

```

We disable most of what we've just defined in the 3042 lines above if we are running in draft mode.

```

3043 \MT@warning@n1{`draft' option active.\MessageBreak
3044 Disabling all micro-typographic extensions.\MessageBreak
3045 This might lead to different line and page breaks}
3046 \MT@protrusionfalse
3047 \MT@expansionfalse

```



```

3048 <*\beta>
3049 \MT@spacingfalse
3050 \MT@kerningfalse
3051 \MT@babelfalse
3052 </\beta>
3053 \let\MT@setupfont\relax
3054 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
3055 \renewcommand*{\UseMicrotypeSet}[2] [] {}
3056 \renewcommand*{\SetProtrusion}[3] [] {}
3057 \renewcommand*{\SetExpansion}[3] [] {}
3058 <*\beta>
3059 \renewcommand*{\SetExtraSpacing}[3] [] {}
3060 \renewcommand*{\SetExtraKerning}[3] [] {}
3061 </\beta>
3062 \renewcommand*{\DeclareCharacterInheritance}[3] [] {}
3063 \renewcommand*{\DeclareMicrotypeAlias}[2] {}
3064 \renewcommand*{\LoadMicrotypeFile}[1] {}
3065 \renewcommand*{\microtypesetup}[1] {}
3066 \renewcommand*{\microtypecontext}[1] {}
3067 \else

```

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

```

3068 \ifnum\pdfoutput<\@ne
3069 \ifMT@opt@expansion \else
3070 \MT@expansionfalse
3071 \fi
3072 \fi

```

pdf<sub>T</sub><sub>X</sub> can create DVI output, too. However, both the DVI viewer and dvips need to find actual fonts. Therefore, expansion will only work if the fonts for different degrees of expansion are readily available.

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

```

3073 \MT@info@n{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3074 \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%

```

**Protrusion.**

```

3075 \ifMT@protrusion
3076 \edef\MT@active@features{\MT@active@features,pr}
3077 \pdfprotrudechars\MT@pr@level
3078 \MT@info@n{Character protrusion enabled (level \number\MT@pr@level)%
3079 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3080 factor: \number\MT@pr@factor\fi
3081 \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`.

```

3082 \MT@ifdefined@c@TF\MT@pr@setname{%
3083 \MT@info@n{Using protrusion set `~\MT@pr@setname'}%
3084 }{%
3085 \global\let\MT@pr@setname\MT@default@pr@set
3086 \MT@info@n{Using default protrusion set `~\MT@pr@setname'}%
3087 }
3088 \else

```

```

3089 \let\MT@protrusion\relax
3090 \MT@info@n1{No character protrusion}
3091 \fi

```

Expansion.

```

3092 \ifMT@expansion

```

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

```

3093 \ifnum\MT@stretch=\m@ne
3094 \let\MT@stretch\MT@stretch@default
3095 \fi

```

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

```

3096 \ifnum\MT@shrink=\m@ne
3097 \ifnum\MT@stretch>\z@
3098 \let\MT@shrink\MT@stretch
3099 \else
3100 \let\MT@shrink\MT@shrink@default
3101 \fi
3102 \fi

```

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

```

3103 \ifnum\MT@step=\m@ne
3104 \ifnum\MT@stretch>\MT@shrink
3105 \ifnum\MT@shrink=\z@
3106 \@tempcnta=\MT@stretch
3107 \else
3108 \@tempcnta=\MT@shrink
3109 \fi
3110 \else
3111 \ifnum\MT@stretch=\z@
3112 \@tempcnta=\MT@shrink
3113 \else
3114 \@tempcnta=\MT@stretch
3115 \fi
3116 \fi
3117 \divide\@tempcnta 5\relax
3118 \else
3119 \@tempcnta=\MT@step
3120 \ifnum\@tempcnta=\z@
3121 \MT@warning@n1{The expansion step cannot be set to zero.\MessageBreak
3122 Setting it to one}
3123 \fi
3124 \fi
3125 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3126 \edef\MT@step{\number\@tempcnta\space}

```

`\MT@auto` Automatic expansion of the font? This new feature of pdf $\text{\TeX}$  1.20 makes the *hz*-algorithm really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdf $\text{\TeX}$ ).

```

3127 \let\MT@auto\@empty
3128 \ifMT@auto
3129 \MT@requires@pdfTeX4{%

```

We turn off automatic expansion if output mode is DVI.

```

3130 \ifnum\pdfoutput<\@ne
3131 \ifMT@opt@auto
3132 \MT@warning@n1{%

```

```

3133         Automatic font expansion only works for PDF output.\MessageBreak
3134         However, you are creating a DVI file. I will switch\MessageBreak
3135         automatic font expansion off and hope that expanded\MessageBreak
3136         fonts are available}
3137     \fi
3138     \MT@autofalse
3139 \else
3140     \def\MT@auto{autoexpand}
3141 \fi

```

Also, if pdf<sub>T</sub><sub>E</sub>X is too old.

```

3142 }{%
3143     \ifMT@opt@auto
3144     \MT@warning@nl{%
3145         The pdftex you are using is too old for automatic\MessageBreak
3146         font expansion. I will switch it off and hope that\MessageBreak
3147         expanded fonts are available on your system.\MessageBreak
3148         Install pdftex version 1.20 or newer}
3149     \fi
3150     \MT@autofalse
3151     \def\MT@auto{1000 }
3152 }

```

No automatic expansion.

```

3153 \else
3154     \ifnum\MT@pdftex@no < 4
3155     \def\MT@auto{1000 }
3156 \fi
3157 \fi

```

Choose the appropriate macro for selected expansion.

```

3158 \ifMT@selected
3159     \let\MT@set@ex@codes\MT@set@ex@codes@s
3160 \else
3161     \let\MT@set@ex@codes\MT@set@ex@codes@n
3162 \fi

```

Filter out stretch=0, shrink=0, since it would result in an pdf<sub>T</sub><sub>E</sub>X error.

```

3163 \ifnum\MT@stretch=\z@
3164     \ifnum\MT@shrink=\z@
3165     \MT@warning@nl{%
3166         Both the stretch and shrink limit are set to zero.\MessageBreak
3167         Disabling font expansion}
3168     \MT@expansionfalse
3169 \fi
3170 \fi
3171 \fi
3172 \ifMT@expansion
3173     \edef\MT@active@features{\MT@active@features,ex}%
3174     \pdfadjustspacing\MT@ex@level
3175     \MT@info@nl{\ifMT@auto\else Non-\fi Automatic font expansion enabled
3176         (level \number\MT@ex@level),\MessageBreak
3177         stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3178         step: \number\MT@step, \ifMT@selected\else non-\fi selected}
3179     \MT@ifdefined@c@TF\MT@ex@setname{%
3180         \MT@info@nl{Using expansion set `~\MT@ex@setname'}%
3181     }{%
3182         \global\let\MT@ex@setname\MT@default@ex@set
3183         \MT@info@nl{Using default expansion set `~\MT@ex@setname'}%
3184     }

```

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

```
3185 \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3186 \color@begingroup\everypar{}\parfillskip\z@skip
3187 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3188 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
```

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

```
3189 \gdef\showhyphens#1{\setbox0\vbox{%
3190 \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3191 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3192 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

3193 \else
3194 \let\MT@expansion\relax
3195 \MT@info@nl{No font expansion}
3196 \fi
3197 \fi
3198 }
3199 <beta>
3200 \MT@requires@pdftex6{
3201 \MT@addto@setup{%
```

Spacing.

```
3202 \ifMT@spacing
3203 \edef\MT@active@features{\MT@active@features,sp}
3204 \pdfadjustinterwordglue\@ne
3205 \MT@info@nl{Adjustment of interword spacing enabled}
3206 \MT@ifdefined@c@TF\MT@sp@setname{%
3207 \MT@info@nl{Using spacing set '\MT@sp@setname'}%
3208 }{%
3209 \global\let\MT@sp@setname\MT@default@sp@set
3210 \MT@info@nl{Using default spacing set '\MT@sp@setname'}%
3211 }
3212 \else
3213 \let\MT@spacing\relax
3214 \MT@info@nl{No adjustment of interword spacing}
3215 \fi
```

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

```
3216 \edef\MT@active@features{\MT@active@features,kn}
3217 \ifMT@kerning
3218 \pdfprependkern\@ne
3219 \pdfappendkern\@ne
3220 \MT@info@nl{Adjustment of character kerning enabled}
3221 \MT@ifdefined@c@TF\MT@kn@setname{%
3222 \MT@info@nl{Using kerning set '\MT@kn@setname'}%
3223 }{%
3224 \global\let\MT@kn@setname\MT@default@kn@set
3225 \MT@info@nl{Using default kerning set '\MT@kn@setname'}%
3226 }
3227 \else
3228 \MT@info@nl{No adjustment of character kerning}
3229 \fi
3230 \ifnum\MT@letterspacing=\m@ne
3231 \let\MT@letterspacing\MT@letterspacing@default
3232 \fi
3233 }
```

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

```
3234 }{
```

```

3235 \MT@addto@setup{%
3236   \ifMT@spacing
3237     \MT@warning@nl{Adjustment of interword spacing only works with\MessageBreak
3238       pdftex version 1.40 or newer. Switching it off}%
3239   \else
3240     \MT@info@nl{No adjustment of interword spacing}
3241   \fi
3242   \MT@spacingfalse
3243   \let\MT@spacing\relax
3244   \ifMT@kerning
3245     \MT@warning@nl{Character kerning only works with\MessageBreak
3246       pdftex version 1.40 or newer. Switching it off}%
3247   \else
3248     \MT@info@nl{No adjustment of character kerning}
3249   \fi
3250   \MT@kerningfalse
3251   \let\MT@kerning\relax
3252 }
3253 }

```

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> on August 16, 2005.

```

3254 \MT@requires@pdftex6{
3255   \AtBeginDocument{%
3256     \ifMT@spacing
3257       \ifMT@babel \else
3258         \ifnum\sfcode`. > 1500
3259           \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
3260             \MT@warning@nl{%
3261               \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3262               interword spacing will disable it. You might want\MessageBreak
3263               to add ` \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
3264               to your preamble}%
3265             }%
3266           \fi
3267         \fi
3268       \fi
3269     }
3270 } \relax
3271 </beta>

```

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

```

3272 \MT@addto@setup{%
3273   \ifx\MT@active@features\@empty \else
3274     \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3275   \fi
3276   \MT@documenttrue
3277 }

```

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

```

3278 <*/beta>
3279 \MT@addto@setup{%
3280   \ifMT@babel
3281     \ifpackageloaded{babel}{%

```

```

3282 \MT@info@nl{Redefining babel's language switching commands}
3283 \let\MT@orig@select@language\select@language
3284 \def\select@language#1{%
3285   \MT@orig@select@language{#1}%
3286   \MT@ifdefined@n@TF{MT@babel@#1}{%
3287     \MT@vinfo{Changing to language `#1' on line \the\inputlineno}%
3288     \expandafter\MT@exp@one@n\expandafter\microtypecontext
3289     \csname MT@babel@#1\endcsname
3290   }{%
3291     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3292   }%
3293 }
3294 \let\MT@orig@foreign@language\foreign@language
3295 \def\foreign@language#1{%
3296   \MT@orig@foreign@language{#1}%
3297   \MT@ifdefined@n@TF{MT@babel@#1}{%
3298     \MT@vinfo{Changing to context `#1' on line \the\inputlineno}%
3299     \expandafter\MT@exp@one@n\expandafter\microtypecontext
3300     \csname MT@babel@#1\endcsname
3301   }{%
3302     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3303   }%
3304 }

```

Disable babel's active characters.

```

3305 \ifMT@kerning
3306   \@tempswafalse
3307   \ifpackagewith{babel}{french}\@tempswatrue\relax
3308   \ifpackagewith{babel}{frenchb}\@tempswatrue\relax
3309   \ifpackagewith{babel}{francais}\@tempswatrue\relax
3310   \if@tempswa
3311     \NoAutoSpaceBeforeFDP
3312     \MT@warning@nl{Switching off French babel's active punctuation characters}%
3313   \fi
3314 \fi

```

In case babel was loaded before microtype:

```

3315 \MT@ifdefined@n@T{MT@babel@\language@name}{%
3316   \MT@vinfo{Changing to context ``\language@name' on line \the\inputlineno}%
3317   \expandafter\MT@exp@one@n\expandafter\microtypecontext
3318   \csname MT@babel@\language@name\endcsname
3319 }{%
3320 }{%
3321   \MT@warning@nl{You did not load the babel package.\MessageBreak
3322     The `babel' option won't have any effect}
3323 }
3324 \fi
3325 }
3326 /beta

```

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.

```

3327 \AtBeginDocument\selectfont

```

\MT@curr@file This is the current file (hopefully with the correct extension).

```

3328 \edef\MT@curr@file{\jobname.tex}

```

That was that.

```

3329 /package

```

## 14 Configuration Files

Let's now write the font configuration files.

```
3330 <*m-t>
3331
```

### 14.1 Font Sets

We first declare some sets in the main configuration file.

```
3332 <*m-t>
3333 %%% -----
3334 %%% FONT SETS
3335
3336 \DeclareMicrotypeSet{all}
3337 { }
3338
3339 \DeclareMicrotypeSet{allmath}
3340 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
3341
3342 \DeclareMicrotypeSet{alltext}
3343 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
3344
3345 \DeclareMicrotypeSet{basicmath}
3346 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
3347   family   = {rm*,sf*},
3348   series    = {md*},
3349   size      = {normalsize,footnotesize,small,large}
3350 }
3351
3352 \DeclareMicrotypeSet{basictext}
3353 { encoding = {OT1,T1,LY1,OT4,QX,T5},
3354   family   = {rm*,sf*},
3355   series    = {md*},
3356   size      = {normalsize,footnotesize,small,large}
3357 }
3358
3359 \DeclareMicrotypeSet{normalfont}
3360 { font = */*/*/*/* }
3361
```

The default sets.

```
3362 %%% -----
3363 %%% DEFAULT SETS
3364
3365 \DeclareMicrotypeSetDefault[protrusion]{alltext}
3366 \DeclareMicrotypeSetDefault[expansion]{basictext}
3367 <*beta>
3368 \DeclareMicrotypeSetDefault[spacing]{basictext}
3369 \DeclareMicrotypeSetDefault[kerning]{alltext}
3370 </beta>
3371
```

### 14.2 Font Aliases

Fonts that are 'the same'.

```
3372 %%% -----
3373 %%% FONT ALIASES
```

3374

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.

```
3375 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
3376 \DeclareMicrotypeAlias{aer}{cmr} % ae
3377 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
3378 \DeclareMicrotypeAlias{cmor}{cmr} % eco
3379 \DeclareMicrotypeAlias{hfor}{cmr} % hfoldsty
```

The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively, also the qfonts package, which provides both font families.

```
3380 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
3381 \DeclareMicrotypeAlias{qpl}{ppl} % qfonts/QuasiPalatino
3382 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
3383 \DeclareMicrotypeAlias{qtm}{ptm} % qfonts/QuasiTimes
```

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

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```
3384 \DeclareMicrotypeAlias{mbch}{bch} % mathdesign/Charter
3385 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

Euro symbol fonts, to save some files.

```
3386 \DeclareMicrotypeAlias{zpeus}{zpeu} % Adobe Euro sans -> serif
3387 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
3388 \DeclareMicrotypeAlias{euroitc}{euroitc} % ITC Euro sans -> serif
3389
```

## 14.3 Interaction with babel

Contexts that are to be set when switching to a language.

```
3390 <*/beta>
3391 %%% -----
3392 %%% INTERACTION WITH THE `babel' PACKAGE
3393
3394 \DeclareMicrotypeBabelHook
3395   {french,français}
3396   {kerning=french, spacing=}
3397
3398 \DeclareMicrotypeBabelHook
3399   {english,american,USenglish,british,UKenglish}
3400   {kerning=, spacing=nonfrench}
3401
3402 \DeclareMicrotypeBabelHook
3403   {turkish}
3404   {kerning=turkish, spacing=}
3405
3406 </beta>
```



## 14.4 Note on Admissible Characters

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

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

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

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

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

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

## 14.5 Character Inheritance

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

```
3407 </m-t>
3408 <*m-t|zpeu|mvs>
3409 %%% -----
3410 %%% CHARACTER INHERITANCE
3411
3412 </m-t|zpeu|mvs>
3413 <*m-t>
```

### 14.5.1 OT1

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

```
3414 \DeclareCharacterInheritance
3415   { encoding = OT1 }
3416   { f = {011}, % ff
3417     i = {\i},
3418     j = {\j},
3419     0 = {\0},
3420     o = {\o},
3421   }
3422
```

### 14.5.2 T1

Candidates here: 028 (‘fi’), 029 (‘fl’), 030 (‘ffi’), 031 (‘ffl’), 156 (‘IJ’ ligature, since  $\LaTeX$  2005/12/01 accessible as \IJ), 188 (‘ij’, \ij),  $\mathcal{A}$ ,  $\mathcal{a}$ ,  $\mathcal{C}$ ,  $\mathcal{c}$ .

```

3423 \DeclareCharacterInheritance
3424 { encoding = T1 }
3425 { A = {\^A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
3426   a = {\^a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
3427   C = {\'C,\c C,\v C},
3428   c = {\'c,\c c,\v c},
3429   D = {\v D,\DH},
3430   d = {\v d,\dj},
3431   E = {\^E,\'E,\^E,\~E,\k E,\v E},
3432   e = {\^e,\'e,\^e,\~e,\k e,\v e},
3433   f = {027}, % ff
3434   G = {\u G},
3435   g = {\u g},
3436   I = {\^I,\'I,\^I,\~I,\"I,\.I},
3437   i = {\^i,\'i,\^i,\~i,\"i,\i},
3438   j = {\j},
3439   L = {\L,\'L,\v L},
3440   l = {\l,\'l,\v l},
3441   N = {\'N,\~N,\v N},
3442   n = {\'n,\~n,\v n},
3443   O = {\O,\^O,\'O,\^O,\~O,\"O,\H O},
3444   o = {\o,\^o,\'o,\^o,\~o,\"o,\H o},
3445   R = {\'R,\v R},
3446   r = {\'r,\v r},
3447   S = {\'S,\c S,\v S,\SS},
3448   s = {\'s,\c s,\v s},
3449   T = {\c T,\v T},
3450   t = {\c t,\v t},
3451   U = {\^U,\'U,\^U,\"U,\H U,\r U},
3452   u = {\^u,\'u,\^u,\"u,\H u,\r u},
3453   Y = {\'Y,\"Y},
3454   y = {\'y,\"y},
3455   Z = {\'Z,\"Z,\v Z},
3456   z = {\'z,\"z,\v z},
3457   - = {127},
3458 }
3459
```

### 14.5.3 LY1

More characters: 008 (‘fl’), 012 (‘fi’), 014 (‘ffi’), 015 (‘ffl’),  $\mathcal{A}$ ,  $\mathcal{a}$ ,  $\mathcal{C}$ ,  $\mathcal{c}$ .

```

3460 \DeclareCharacterInheritance
3461 { encoding = LY1 }
3462 { A = {\^A,\'A,\^A,\~A,\"A,\r A},
3463   a = {\^a,\'a,\^a,\~a,\"a,\r a},
3464   C = {\c C},
3465   c = {\c c},
3466   D = {\DH},
3467   E = {\^E,\'E,\^E,\~E},
3468   e = {\^e,\'e,\^e,\~e},
3469   f = {011}, % ff
3470   I = {\^I,\'I,\^I,\"I},
3471   i = {\^i,\'i,\^i,\"i,\i},
3472   L = {\L},
3473   l = {\l},

```

```

3474     N = {\~N},
3475     n = {\~n},
3476     O = {\~0,\'0,\^0,\~0,\"0,\0},
3477     o = {\~o,\'o,\^o,\~o,\"o,\o},
3478     S = {\v S},
3479     s = {\v s},
3480     U = {\~U,\'U,\^U,\"U},
3481     u = {\~u,\'u,\^u,\"u},
3482     Y = {\'Y,\"Y},
3483     y = {\'y,\"y},
3484     Z = {\v Z},
3485     z = {\v z},
3486   }
3487

```

#### 14.5.4 OT4

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

```

3488 \DeclareCharacterInheritance
3489   { encoding = OT4 }
3490   { A = {\k A},
3491     a = {\k a},
3492     C = {\'C},
3493     c = {\'c},
3494     E = {\k E},
3495     e = {\k e},
3496     f = {011}, % ff
3497     i = {\i},
3498     j = {\j},
3499     L = {\L},
3500     l = {\l},
3501     N = {\'N},
3502     n = {\'n},
3503     O = {\0,\'0},
3504     o = {\o,\'o},
3505     S = {\'S},
3506     s = {\'s},
3507     Z = {\'Z,\.Z},
3508     z = {\'z,\.z},
3509   }
3510

```

#### 14.5.5 QX

The Central European QX encoding. Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

3511 \DeclareCharacterInheritance
3512   { encoding = QX }
3513   { A = {\~A,\'A,\^A,\~A,\"A,\k A,\AA},
3514     a = {\~a,\'a,\^a,\~a,\"a,\k a,\aa},
3515     C = {\'C,\c C},
3516     c = {\'c,\c c},
3517     D = {\DH},
3518     E = {\~E,\'E,\^E,\"E,\k E},
3519     e = {\~e,\'e,\^e,\"e,\k e},
3520     f = {011}, % ff
3521     I = {\~I,\'I,\^I,\"I,\k I},

```

```

3522 i = {\~i,\'i,\^i,\"i,\k i,\i},
3523 j = {\j},
3524 L = {\L},
3525 l = {\l},
3526 N = {\'N,\-N},
3527 n = {\'n,\-n},
3528 O = {\0,\`0,\'0,\^0,\~0,\"0},
3529 o = {\o,\`o,\'o,\^o,\~o,\"o},
3530 S = {\'S,\c S,\v S},
3531 s = {\'s,\c s,\v s},
3532 T = {\c T},
3533 t = {\c t},
3534 U = {\~U,\'U,\^U,\"U,\k U},
3535 u = {\~u,\'u,\^u,\"u,\k u},
3536 Y = {\'Y,\"Y},
3537 y = {\'y,\"y},
3538 Z = {\'Z,\.Z,\v Z},
3539 z = {\'z,\.z,\v z},
3540 . = \textellipsis
3541 }
3542

```

### 14.5.6 T5

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

```

3543 \DeclareCharacterInheritance
3544 { encoding = T5 }
3545 { A = {\^A,\'A,\-A,\h A,\d A,\^A,\u A,
3546       \~\Acircumflex,\'\Acircumflex,\-\Acircumflex,\h\Acircumflex,\d\Acircumflex,
3547       \~\Abreve,\'\Abreve,\-\Abreve,\h\Abreve,\d\Abreve},
3548   a = {\^a,\'a,\-a,\h a,\d a,\^a,\u a,
3549       \~\acircumflex,\'\acircumflex,\-\acircumflex,\h\acircumflex,\d\acircumflex,
3550       \~\abreve,\'\abreve,\-\abreve,\h\abreve,\d\abreve},
3551   D = {\DJ},
3552   d = {\dj},
3553   E = {\^E,\'E,\-E,\h E,\d E,\^E,
3554       \~\Ecircumflex,\'\Ecircumflex,\-\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
3555   e = {\^e,\'e,\-e,\h e,\d e,\^e,
3556       \~\ecircumflex,\'\ecircumflex,\-\ecircumflex,\h\ecircumflex,\d\ecircumflex},
3557   I = {\^I,\'I,\-I,\h I,\d I},
3558   i = {\~i,\'i,\-i,\h i,\d i,\i},
3559   O = {\^O,\'O,\-O,\h O,\d O,\^O,\horn O,
3560       \~\Ocircumflex,\'\Ocircumflex,\-\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
3561       \~\Ohorn,\'\Ohorn,\-\Ohorn,\h\Ohorn,\d\Ohorn},
3562   o = {\^o,\'o,\-o,\h o,\d o,\^o,\horn o,
3563       \~\ocircumflex,\'\ocircumflex,\-\ocircumflex,\h\ocircumflex,\d\ocircumflex,
3564       \~\ohorn,\'\ohorn,\-\ohorn,\h\ohorn,\d\ohorn},
3565   U = {\^U,\'U,\-U,\h U,\d U,\horn U,
3566       \~\Uhorn,\'\Uhorn,\-\Uhorn,\h\Uhorn,\d\Uhorn},
3567   u = {\~u,\'u,\-u,\h u,\d u,\horn u,
3568       \~\uhorn,\'\uhorn,\-\uhorn,\h\uhorn,\d\uhorn},
3569   Y = {\^Y,\'Y,\-Y,\h Y,\d Y},
3570   y = {\~y,\'y,\-y,\h y,\d y},
3571 }
3572
3573 </m-t>

```

### 14.5.7 Euro symbols

Make Euro symbols settings simpler.

```

3574 < *zpeu >
3575 \DeclareCharacterInheritance
3576   { encoding = U,
3577     family   = {zpeu,zpeus,eurosans} }
3578   {
3579     E = 128
3580   }
3581
3582 < /zpeu >
3583 < *mvs >
3584 \DeclareCharacterInheritance
3585   { encoding = OT1,
3586     family   = mvs }
3587   {
3588     164 = {099,100,101} % \EURhv,\EURcr,\EURtm
3589   }
3590
3591 < /mvs >
3592 < *m-t >

```

## 14.6 Font Expansion

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

```

3593 %%% -----
3594 %%% EXPANSION SETTINGS
3595
3596 \SetExpansion
3597   [ name      = default      ]
3598   { encoding = {OT1,OT4,QX,T1,LY1} }
3599   {
3600     A = 500,      a = 700,
3601     \AE = 500,    \ae = 700,
3602     B = 700,      b = 700,
3603     C = 700,      c = 700,
3604     D = 500,      d = 700,
3605     E = 700,      e = 700,
3606     F = 700,
3607     G = 500,      g = 700,
3608     H = 700,      h = 700,
3609     K = 700,      k = 700,
3610     M = 700,      m = 700,
3611     N = 700,      n = 700,
3612     O = 500,      o = 700,
3613     \OE = 500,    \oe = 700,
3614     P = 700,      p = 700,
3615     Q = 500,      q = 700,
3616     R = 700,
3617     S = 700,      s = 700,
3618     U = 700,      u = 700,
3619     W = 700,      w = 700,
3620     Z = 700,      z = 700,
3621     2 = 700,
3622     3 = 700,
3623     6 = 700,
3624     8 = 700,

```

```

3625     9 = 700,
3626   }
3627

```

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

```

3628 \SetExpansion
3629   [ name      = T5 ]
3630   { encoding = T5 }
3631   {
3632     A = 500,      a = 700,
3633     B = 700,      b = 700,
3634     C = 700,      c = 700,
3635     D = 500,      d = 700,
3636     E = 700,      e = 700,
3637     F = 700,
3638     G = 500,      g = 700,
3639     H = 700,      h = 700,
3640     K = 700,      k = 700,
3641     M = 700,      m = 700,
3642     N = 700,      n = 700,
3643     O = 500,      o = 700,
3644     P = 700,      p = 700,
3645     Q = 500,      q = 700,
3646     R = 700,
3647     S = 700,      s = 700,
3648     U = 700,      u = 700,
3649     W = 700,      w = 700,
3650     Z = 700,      z = 700,
3651     2 = 700,
3652     3 = 700,
3653     6 = 700,
3654     8 = 700,
3655     9 = 700,
3656   }
3657
3658 </m-t>

```

## 14.7 Character Protrusion

```

3659 %%% -----
3660 %%% PROTRUSION SETTINGS
3661

```

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},    \textendash      = { ,200},
\textquoteright = {700, },    \textquoteright = { ,700},
\textquotedblleft = {500, },    \textquotedblright = { ,500},
}

```

### 14.7.1 Default

The default settings always use the most moderate value.

```

3662 <*cfg-t>
3663 \SetProtrusion
3664 <m-t> [ name = default ]

We also create configuration files for the fonts Bitstream Charter (NFSS code bch),
3665 <bch> [ name = bch-default ]

Computer Modern Roman (cmr),
3666 <cmr> [ name = cmr-default ]

Adobe Garamond (pad, padx, padj),
3667 <pad> [ name = pad-default ]

Minion15 (pmnx, pmnj),
3668 <pmn> [ name = pmnj-default ]

Palatino (ppl, pplx, pplj),
3669 <ppl> [ name = ppl-default ]

Times (ptm, ptmx, ptmj),
3670 <ptm> [ name = ptm-default ]

and URW Garamond (ugm).
3671 <ugm> [ name = ugm-default ]
3672 <m-t> { encoding = OT1 }
3673 <cmr> { }
3674 <bch|pad|pmn|ugm> { encoding = OT1,
3675 <ppl|ptm> { encoding = {OT1,OT4},
3676 <bch> family = bch }
3677 <pad> family = {pad,padx,padj} }
3678 <pmn> family = pmnj }
3679 <ppl> family = {ppl,pplx,pplj} }
3680 <ptm> family = {ptm,ptmx,ptmj} }
3681 <ugm> family = ugm }
3682 {
3683 <m-t|bch|cmr|pad|pmn|ppl|ptm> A = {50,50},
3684 <ugm> A = {50,100},
3685 <m-t|pad|ptm> \AE = {50, },
3686 <ugm> \AE = {150,50},
3687 <ugm> B = { ,50},

```

15 Contributed by Harald Harders ([h.harders@tu-bs.de](mailto:h.harders@tu-bs.de))

```

3688 <bch|pad|pmn|ugm>      C = {50,  },
3689 <bch|pad|pmn>          D = {  ,50},
3690 <ugm>                  D = {  ,70},
3691 <ugm>                  E = {  ,50},
3692 <m-t|bch|cmr|pad|pmn|ptm>      F = {  ,50},
3693 <ugm>                  F = {  ,70},
3694 <bch|pad|pmn>          G = {50,  },
3695 <ugm>                  G = {50,50},
3696 <m-t|cmr|pad|pmn|ppl|ptm|ugm>      J = {50,  },
3697 <bch>                  J = {100,  },
3698                      K = {  ,50},
3699 <m-t|bch|cmr|pad|pmn|ppl>      L = {  ,50},
3700 <ptm>                  L = {  ,80},
3701 <ugm>                  L = {  ,120},
3702 <bch|pad|pmn|ugm>          O = {50,50},
3703 <pad|pmn>              \OE = {50,  },
3704 <ugm>                  \OE = {50,50},
3705 <ugm>                  P = {  ,50},
3706 <bch|pad|pmn>          Q = {50,70},
3707 <ugm>                  Q = {50,50},
3708 <bch>                  R = {  ,50},
3709 <ugm>                  R = {  ,70},
3710 <m-t|bch|cmr|pad|pmn|ppl|ptm>      T = {50,50},
3711 <ugm>                  T = {70,70},
3712 <m-t|bch|cmr|pad|pmn|ppl|ptm>      V = {50,50},
3713 <ugm>                  V = {70,70},
3714 <m-t|bch|cmr|pad|pmn|ppl|ptm>      W = {50,50},
3715 <ugm>                  W = {70,70},
3716 <m-t|bch|cmr|pad|pmn|ppl|ptm>      X = {50,50},
3717 <ugm>                  X = {50,70},
3718 <m-t|bch|cmr|pad|pmn|ppl>          Y = {50,50},
3719 <ptm|ugm>              Y = {80,80},
3720 <ugm>                  Z = {50,50},
3721 <m-t|bch|cmr|pad|pmn|ppl|ptm>      k = {  ,50},
3722 <ugm>                  k = {  ,70},
3723 <pmn>                  l = {  ,-50},
3724 <pad|ppl>              p = {50,50},
3725 <ugm>                  p = {  ,50},
3726 <pad|ppl>              q = {50,  },
3727                      r = {  ,50},
3728 <cmr|pad|pmn>          t = {  ,70},
3729 <bch>                  t = {  ,50},
3730 <ugm>                  t = {  ,100},
3731 <m-t|bch|cmr|pad|pmn|ppl|ptm>      v = {50,50},
3732 <ugm>                  v = {50,70},
3733 <m-t|bch|cmr|pad|pmn|ppl|ptm>      w = {50,50},
3734 <ugm>                  w = {50,70},
3735                      x = {50,50},
3736 <m-t|bch|pad|pmn>          y = {  ,50},
3737 <cmr|ppl|ptm>          y = {50,70},
3738 <ugm>                  y = {  ,70},

3739 <cmr>                  0 = {  ,50},
3740 <m-t>                  1 = {50,50},
3741 <bch|pad|ptm|ugm>          1 = {150,150},
3742 <cmr>                  1 = {100,200},
3743 <pmn>                  1 = {  ,50},
3744 <ppl>                  1 = {100,100},
3745 <bch|cmr|pad|ugm>          2 = {50,50},
3746 <cmr|pad|ugm>          3 = {50,50},
3747 <bch|pmn>              3 = {50,  },
3748 <m-t|pad>              4 = {50,50},

```



```

3749 <bch> 4 = {100,50},
3750 <cmr|ugm> 4 = {70,70},
3751 <pmn> 4 = {50, },
3752 <ptm> 4 = {70, },
3753 <cmr> 5 = { ,50},
3754 <pad> 5 = {50,50},
3755 <bch> 6 = {50, },
3756 <cmr> 6 = { ,50},
3757 <pad> 6 = {50,50},
3758 <m-t> 7 = {50,50},
3759 <bch|pad|pmn|ugm> 7 = {50,80},
3760 <cmr|ptm> 7 = {50,100},
3761 <ppl> 7 = { ,50},
3762 <cmr> 8 = { ,50},
3763 <bch|pad> 9 = {50,50},
3764 <cmr> 9 = { ,50},
3765 <m-t|cmr|pad|pmn|ppl|ptm|ugm> . = { ,700},
3766 <bch> . = { ,600},
3767 {,}= { ,500},
3768 <m-t|cmr|pad|pmn|ppl|ptm|ugm> : = { ,500},
3769 <bch> : = { ,400},
3770 <m-t|bch|pad|pmn|ptm> ; = { ,300},
3771 <cmr|ppl> ; = { ,500},
3772 <ugm> ; = { ,400},
3773 ! = { ,100},
3774 <m-t|pad|pmn|ptm> ? = { ,100},
3775 <bch|cmr|ppl|ugm> ? = { ,200},
3776 <pmn> " = {300,300},
3777 <m-t|bch|cmr|pad|pmn|ppl> @ = {50,50},
3778 <ptm> @ = {100,100},
3779 <m-t|bch|cmr|pad|pmn|ppl|ptm> ~ = {200,250},
3780 <ugm> ~ = {300,350},
3781 <pad|ppl|ptm> & = {50,100},
3782 <ugm> & = { ,100},
3783 <m-t|cmr|pad|pmn> \% = {50,50},
3784 <bch> \% = { ,50},
3785 <ppl|ptm> \% = {100,100},
3786 <ugm> \% = {50,100},
3787 <m-t|ppl|ptm|ugm> * = {200,200},
3788 <bch|pmn> * = {200,300},
3789 <cmr|pad> * = {300,300},
3790 <m-t|cmr|ppl|ptm> + = {250,250},
3791 <bch> + = {150,250},
3792 <pad> + = {300,300},
3793 <pmn> + = {150,200},
3794 <ugm> + = {250,300},
3795 <ugm> {=}= {200,200},
3796 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
3797 <bch|ugm> ( = {200, }, ) = { ,200},
3798 <cmr|ppl> ( = {100, }, ) = { ,300},
3799 <bch|pmn> [ = {100, }, ] = { ,100},

3800 <m-t|pad|pmn|ptm> / = {100,200},
3801 <bch> / = { ,200},
3802 <cmr|ppl> / = {200,300},
3803 <ugm> / = {100,300},
3804 <m-t|ptm> - = {500,500},
3805 <bch|cmr|ppl> - = {400,500},
3806 <pad> - = {300,500},
3807 <pmn> - = {200,400},
3808 <ugm> - = {500,600},

```

```

3809 <m-t|pmn> \textendash = {200,200}, \textemdash = {150,150},
3810 <bch> \textendash = {200,300}, \textemdash = {150,250},
3811 <cmr> \textendash = {400,300}, \textemdash = {300,200},
3812 <pad|ppl|ptm> \textendash = {300,300}, \textemdash = {200,200},
3813 <ugm> \textendash = {250,300}, \textemdash = {250,250},

```

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

```

3814 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
3815 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
3816 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
3817 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
3818 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
3819 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300},
3820 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600},
3821 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400},
3822 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400},
3823 }
3824

```

Greek uppercase letters are in OT1 encoding only.

```

3825 <*cmr>
3826 \SetProtrusion
3827 [ name = cmr-OT1,
3828   load = cmr-default ]
3829 { encoding = {OT1,OT4},
3830   family = cmr }
3831 {
3832   \AE = { 50, },
3833   "00 = { ,150}, % \Gamma
3834   "01 = {100,100}, % \Delta
3835   "02 = { 50, 50}, % \Theta
3836   "03 = {100,100}, % \Lambda
3837 % "04 = { , }, % \Xi
3838 % "05 = { , }, % \Pi
3839   "06 = { 50, 50}, % \Sigma
3840   "07 = {100,100}, % \Upsilon
3841   "08 = { 50, 50}, % \Phi
3842   "09 = { 50, 50}, % \Psi
3843 % "0A = { , }, % \Omega
3844 }
3845
3846 </cmr>

```

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

```

3847 \SetProtrusion
3848 <m-t> [ name = T1-default,
3849 <bch> [ name = bch-T1,
3850 <cmr> [ name = cmr-T1,
3851 <pad> [ name = pad-T1,
3852 <pmn> [ name = pmnj-T1,
3853 <ppl> [ name = ppl-T1,
3854 <ptm> [ name = ptm-T1,
3855 <ugm> [ name = ugm-T1,
3856 <m-t> load = default ]
3857 <bch> load = bch-default ]
3858 <cmr> load = cmr-default ]
3859 <pad> load = pad-default ]
3860 <pmn> load = pmnj-default ]
3861 <ppl> load = ppl-default ]

```

```

3862 <ptm>    load    = ptm-default ]
3863 <ugm>    load    = ugm-default ]
3864 <m-t>    { encoding = {T1,LY1}    }
3865 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
3866 <ptm|ugm> { encoding = {T1},
3867 <bch>     family   = bch }
3868 <cmr>     family   = cmr }
3869 <pad>     family   = {pad,padx,padj} }
3870 <pmn>     family   = pmnj }
3871 <ppl>     family   = {ppl,pplx,pplj} }
3872 <ptm>     family   = {ptm,ptmx,ptmj} }
3873 <ugm>     family   = ugm }
3874 {
3875 <cmr>     \AE = {50, },
3876 <bch>     \OE = {50, },
3877 <pmn>     \TH = { ,50},
3878 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
3879 <cmr>     _ = {200,200},
3880 <ugm>     _ = {100,200},
3881 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
3882 <bch>     \textbackslash = {150,200},
3883 <cmr|ppl> \textbackslash = {200,300},
3884 <ugm>     \textbackslash = {100,300},
3885 <ugm>     \textbar    = {200,200},
3886 <cmr>     \textquotedblleft = {200,600},
3887 <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.

```

3888 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3889 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
3890 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsingright = {300,400},
3891 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsingright = {300,500},
3892 <ugm> \guilsinglleft = {400,400}, \guilsingright = {300,600},
3893 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
3894 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
3895 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
3896 <pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
3897 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
3898 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
3899 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
3900 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
3901 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
3902 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200},
3903 <pmn> \textless = {100, }, \textgreater = { ,100},
3904 <pmn> \textvisiblespace = {100,100}, % not in LY1
3905 % \dh = { , },
3906 % \th = { , },
3907 % \NG = { , },
3908 % \ng = { , },
3909 % \textasciicircum = { , },
3910 % \textsterling = { , }, % also in TS1
3911 % \textsection = { , }, % also in TS1
3912 }
3913

```

The lmodern fonts, on the other hand, restore the original kerning from the OT1 fonts, and so do we. Silly, isn't it?

```

3914 <*cmr>
3915 \SetProtrusion
3916 [ name = lmr-T1,

```

```

3917     load      = cmr-T1 ]
3918     { encoding = {T1,LY1},
3919       family   = lmr      }
3920     {
3921       \textquotedblleft = {500,300},
3922       \quotedblbase     = {500,300},
3923     }
3924
3925 </cmr>

```

Settings for the QX encoding (generic and Times). It also includes some glyphs otherwise in TS1.

```

3926 <*-t|ptm>
3927 \SetProtrusion
3928 <m-t> [ name      = QX-default,
3929 <ptm> [ name      = ptm-QX,
3930 <m-t>   load      = default ]
3931 <ptm>   load      = ptm-default ]
3932 <m-t>   { encoding = QX }
3933 <ptm>   { encoding = QX,
3934 <ptm>     family   = {ptm,ptmx,ptmj} }
3935     {
3936 <ptm>       * = {200,200},
3937     {=} = {100,100},
3938     \textunderscore = {100,100},
3939     \textbackslash    = {100,200},
3940     \quotedblbase    = {400,400},
3941 <m-t>     \guillemotleft = {200,200}, \guillemotright = {200,200},
3942 <ptm>     \guillemotleft = {300,300}, \guillemotright = {200,400},
3943     \textexclamdown = {100, }, \textquestiondown = {100, },
3944 <m-t>     \textbraceleft = {400,200}, \textbraceright = {200,400},
3945 <ptm>     \textbraceleft = {200,200}, \textbraceright = {200,300},
3946     \textless        = {200,100}, \textgreater        = {100,200},
3947     \textminus       = {200,200}, \textdegree       = {300,300},
3948 <m-t>     \copyright    = {100,100}, \textregistered = {100,100},
3949 <ptm>     \copyright    = {100,150}, \textregistered = {100,150},
3950 <ptm>     \textxgeq     = { ,100}, \textxleq      = {100, },
3951 <ptm>     \textalpha    = { , 50}, \textDelta     = { 70, 70},
3952 <ptm>     \textpi       = { 50, 80}, \textSigma     = { , 70},
3953 <ptm>     \textmu       = { , 80}, \texteuro       = { 50, 50},
3954 <ptm>     \textellipsis = {175,175}, \textasciitilde = { 80, 80},
3955 <ptm>     \textapprox   = { 50, 50}, \textinfty     = {100,100},
3956 <ptm>     \textdagger   = {150,150}, \textdaggerdbl  = {100,100},
3957 <ptm>     \textdiv      = { 50,150}, \textsection    = { 80, 80},
3958 <ptm>     \texttimes    = {100,150}, \textpm         = { 50, 80},
3959 <ptm>     \textbullet   = {150,150}, \textperiodcentered = {300,300},
3960 <ptm>     \textquotesingle = {500,500}, \textquotedbl = {300,300},
3961 <ptm>     \textperthousand = { ,50},
3962     }
3963
3964 </m-t|ptm>
3965 <*-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.

```

3966 \SetProtrusion
3967 <cmr> [ name      = cmr-T5,
3968 <cmr>   load      = cmr-default ]
3969 <bch> [ name      = bch-T5,
3970 <bch>   load      = bch-default ]
3971     { encoding = T5,

```

```

3972 <cmr>    family = cmr }
3973 <bch>    family = bch }
3974 {
3975 <bch>    _ = {100,100},
3976 <bch>    \textbackslash = {150,200},
3977 <cmr>    \textbackslash = {200,300},
3978 <cmr>    \textquotedblleft = {200,600},
3979 <cmr>    \textquotedbl = {300,300},
3980 <bch>    \quotesinglbase = {400,400}, \quotedblbase = {300,300},
3981 <cmr>    \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3982 <bch>    \guilsinglleft = {400,300}, \guilsinglright = {300,400},
3983 <cmr>    \guilsinglleft = {400,400}, \guilsinglright = {300,500},
3984 <bch>    \guillemotleft = {200,200}, \guillemotright = {150,300},
3985 <cmr>    \guillemotleft = {300,200}, \guillemotright = {100,400},
3986 <bch>    \textbraceleft = {200, }, \textbraceright = { ,300},
3987 <cmr>    \textbraceleft = {400,200}, \textbraceright = {200,400},
3988 \textless = {200,100}, \textgreater = {100,200},
3989 }
3990
3991 </cmr|bch>
3992 <*pmn>
3993 \SetProtrusion
3994 [ name = pmnx-OT1,
3995   load = pmnj-default ]
3996 { encoding = OT1,
3997   family = pmnx }
3998 {
3999   1 = {230,180},
4000 }
4001
4002 \SetProtrusion
4003 [ name = pmnx-T1,
4004   load = pmnj-T1 ]
4005 { encoding = {T1,LY1},
4006   family = pmnx }
4007 {
4008   1 = {230,180},
4009 }
4010
4011 </pmn>

```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```

4012 <*ptm>
4013 \SetProtrusion
4014 [ name = ptm-LY1,
4015   load = ptm-T1 ]
4016 { encoding = LY1,
4017   family = {ptm,ptmx,ptmj} }
4018 {
4019   _ = {100,100},
4020   \texttrademark = {100,100},
4021   \textregistered = {100,100},
4022   \textcopyright = {100,100},
4023   \textdegree = {300,300},
4024   \textminus = {200,200},
4025   \textellipsis = {100,100},
4026   \texteuro = { , }, % ?
4027   \textcent = {100,100},
4028   \textquotesingle = {500,500},
4029   \textflorin = { 50, 70},

```

```

4030 \textdagger           = {150,150},
4031 \textdaggerdbl        = {100,100},
4032 \textperthousand      = { , 50},
4033 \textbullet           = {150,150},
4034 \textonesuperior      = {100,100},
4035 \texttwosuperior      = { 50, 50},
4036 \textthreesuperior    = { 50, 50},
4037 \textperiodcentered  = {300,300},
4038 \textplusminus        = { 50, 80},
4039 \textmultiply         = {100,100},
4040 \textdivide           = { 50,150},
4041 % \textbrokenbar      = { , },
4042 % \textyen            = { , },
4043 % \textfractionsolidus = { , },
4044 % \textordfeminine    = { , },
4045 % \textordmasculine   = { , },
4046 % \textmu             = { , },
4047 % \textparagraph     = { , },
4048 % \textonequarter     = { , },
4049 % \textonehalf        = { , },
4050 % \textthreequarters  = { , },
4051 }
4052
4053 </ptm>

```

### 14.7.2 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.

```

4054 \SetProtrusion
4055 <m-t> [ name = OT1-it ]
4056 <bch> [ name = bch-it ]
4057 <cmr> [ name = cmr-it ]
4058 <pad> [ name = pad-it ]
4059 <pmn> [ name = pmn-j-it ]
4060 <ppl> [ name = ppl-it ]
4061 <ptm> [ name = ptm-it ]
4062 <ugm> [ name = ugm-it ]
4063 <m-t|bch|pad|pmn|ugm> { encoding = OT1,
4064 <ppl|ptm> { encoding = {OT1,OT4},
4065 <bch> family = bch,
4066 <pad> family = {pad,padx,padj},
4067 <pmn> family = pmnj,
4068 <ppl> family = {ppl,pplx,pplj},
4069 <ptm> family = {ptm,ptmx,ptmj},
4070 <ugm> family = ugm,
4071 <!(cmr|ugm)> shape = {it,sl} }
4072 <ugm> shape = it }
4073 <cmr> { }
4074 {
4075 <cmr|ptm> A = {100,50},
4076 <pad|pmn> A = {50, },
4077 <ugm> A = { ,150},
4078 <ppl> A = {50,50},
4079 <ptm> \AE = {100, },
4080 <pad|ppl> \AE = {50, },
4081 <pmn> \AE = { , -50},
4082 <cmr|pad|ppl|ptm> B = {50, },

```

```

4083 <pmn>      B = {20,-50},
4084 <bch|ppl|ptm|ugm> C = {50, },
4085 <cmr|pad>    C = {100, },
4086 <pmn>      C = {50,-50},
4087 <cmr|pad|ppl|ptm> D = {50,50},
4088 <pmn>      D = {20, },
4089 <cmr|pad|ppl|ptm> E = {50, },
4090 <pmn>      E = {20,-50},
4091 <cmr|pad|ptm> F = {100, },
4092 <pmn>      F = {10, },
4093 <ppl>      F = {50, },
4094 <bch|ppl|ptm|ugm> G = {50, },
4095 <cmr|pad>    G = {100, },
4096 <pmn>      G = {50,-50},
4097 <cmr|pad|ppl|ptm> H = {50, },
4098 <cmr|pad|ptm> I = {50, },
4099 <pmn>      I = {20,-50},
4100 <cmr|ptm>   J = {100, },
4101 <pad>      J = {50, },
4102 <pmn>      J = {20, },
4103 <cmr|pad|ppl|ptm> K = {50, },
4104 <pmn>      K = {20, },
4105 <cmr|pad|ppl|ptm> L = {50, },
4106 <pmn>      L = {20,50},
4107 <ugm>      L = { ,100},
4108 <cmr|ptm>   M = {50, },
4109 <pmn>      M = { , -30},
4110 <cmr|ptm>   N = {50, },
4111 <pmn>      N = { , -30},
4112 <bch|pmn|ppl|ptm> O = {50, },
4113 <cmr|pad>   O = {100, },
4114 <ugm>      O = {70,50},
4115 <pmn|ppl|ptm> \OE = {50, },
4116 <pad>      \OE = {100, },
4117 <cmr|pad|ppl|ptm> P = {50, },
4118 <pmn>      P = {20,-50},
4119 <bch|pmn|ppl|ptm> Q = {50, },
4120 <cmr|pad>   Q = {100, },
4121 <ugm>      Q = {70,50},
4122 <cmr|pad|ppl|ptm> R = {50, },
4123 <pmn>      R = {20, },
4124 <bch|cmr|pad|ppl|ptm> S = {50, },
4125 <pmn>      S = {20,-30},
4126 <bch|cmr|pad|ppl|ptm> $ = {50, },
4127 <pmn>      $ = {20,-30},
4128 <bch|pmn|ugm> T = {70, },
4129 <cmr|pad|ppl|ptm> T = {100, },
4130 <cmr|pad|ppl|ptm> U = {50, },
4131 <pmn>      U = {50,-50},
4132 <cmr|pad|pmn|ugm> V = {100, },
4133 <ppl|ptm>   V = {100,50},
4134 <cmr|pad|pmn|ugm> W = {100, },
4135 <ppl>      W = {50, },
4136 <ptm>      W = {100,50},
4137 <cmr|ppl|ptm> X = {50, },
4138 <cmr|ptm>   Y = {100, },
4139 <pmn>      Y = {50, },
4140 <ppl>      Y = {100,50},
4141 <pmn>      Z = { , -50},
4142 <pmn>      d = { , -50},
4143 <pad|pmn>   f = { , -100},

```

```

4144 <pmn> i = { , -30},
4145 <pmn> j = { , -30},
4146 <pmn> l = { , -100},
4147 <bch> o = {50, 50},
4148 <bch> p = { , 50},
4149 <pmn> p = { -50, },
4150 <bch> q = {50, },
4151 <pmn> r = { , 50},
4152 <bch> t = { , 50},
4153 <pmn|ugm> v = {50, },
4154 <bch> w = { , 50},
4155 <pmn|ugm> w = {50, },
4156 <bch> y = { , 50},
4157 <cmr> 0 = {100, },
4158 <bch|ptm> 1 = {150, 100},
4159 <cmr> 1 = {200, 50},
4160 <pad> 1 = {150, },
4161 <pmn> 1 = {50, },
4162 <ppl> 1 = {100, },
4163 <ugm> 1 = {150, 150},
4164 <cmr> 2 = {100, -100},
4165 <pad|ppl|ptm> 2 = {50, },
4166 <pmn> 2 = { -50, },
4167 <bch> 3 = {50, },
4168 <cmr> 3 = {100, -100},
4169 <pmn> 3 = { -100, },
4170 <ptm> 3 = {100, 50},
4171 <bch> 4 = {100, },
4172 <cmr|pad> 4 = {150, },
4173 <ppl|ptm> 4 = {50, },
4174 <cmr> 5 = {100, },
4175 <ptm> 5 = {50, },
4176 <bch> 6 = {50, },
4177 <cmr> 6 = {100, },
4178 <bch|pad|ptm> 7 = {100, },
4179 <cmr> 7 = {200, -150},
4180 <pmn> 7 = {20, },
4181 <ppl> 7 = {50, },
4182 <cmr> 8 = {50, -50},
4183 <cmr> 9 = {100, -100},
4184 <m-t|cmr|pad|pmn|ppl> . = { , 500},
4185 <bch|ptm|ugm> . = { , 700},
4186 <m-t|cmr|pad|pmn|ppl> {,}= { , 500},
4187 <bch|ugm> {,}= { , 600},
4188 <ptm> {,}= { , 700},
4189 <m-t|cmr|pad|ppl> : = { , 300},
4190 <bch|ugm> : = { , 400},
4191 <pmn> : = { , 200},
4192 <ptm> : = { , 500},
4193 <m-t|cmr|pad|ppl> ; = { , 300},
4194 <bch|ugm> ; = { , 400},
4195 <pmn> ; = { , 200},
4196 <ptm> ; = { , 500},
4197 <ptm> ! = { , 100},
4198 <bch> ? = { , 200},
4199 <ptm> ? = { , 100},
4200 <ppl> ? = { , 300},
4201 <pmn> " = {400, 200},
4202 <m-t|pad|pmn|ppl|ptm> & = {50, 50},
4203 <bch> & = { , 80},
4204 <cmr> & = {100, 50},

```



```

4205 <ugm>      & = {50,100},
4206 <m-t|cmr|pad|pmn> \% = {100, },
4207 <bch>      \% = {50,50},
4208 <ppl|ptm>   \% = {100,100},
4209 <ugm>      \% = {100,50},
4210 <m-t|pmn|ppl> * = {200,200},
4211 <bch>      * = {300,200},
4212 <cmr>      * = {400,100},
4213 <pad>      * = {500,100},
4214 <ptm|ugm>  * = {400,200},
4215 <m-t|cmr|pmn|ppl> + = {150,200},
4216 <bch|ugm>  + = {250,250},
4217 <pad|ptm>  + = {250,200},
4218 <m-t|pad|pmn|ppl> @ = {50,50},
4219 <bch>      @ = {80,50},
4220 <cmr>      @ = {200,50},
4221 <ptm>      @ = {150,150},
4222 <m-t|bch|ugm> ~ = {150,150},
4223 <cmr|pad|pmn|ppl|ptm> - = {200,150},
4224 <ugm>      {=} = {200,200},
4225 ( = {200, }, ) = { ,200},
4226 <m-t|cmr|pad|ppl|ptm|ugm> / = {100,200},
4227 <bch>      / = { ,150},
4228 <pmn>      / = {100,150},
4229 <m-t>      - = {300,300},
4230 <bch|pad>  - = {300,400},
4231 <pmn>      - = {200,300},
4232 <cmr>      - = {500,300},
4233 <ppl>      - = {300,500},
4234 <ptm>      - = {500,500},
4235 <ugm>      - = {400,700},
4236 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
4237 <bch>      \textendash = {200,300}, \textendash = {150,200},
4238 <cmr>      \textendash = {500,300}, \textendash = {400,200},
4239 <pad|ppl|ptm|ugm> \textendash = {300,300}, \textendash = {200,200},
4240 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
4241 <cmr|pad> \textquoteleft = {800,200}, \textquoteright = {800,200},
4242 <ppl>      \textquoteleft = {700,400}, \textquoteright = {700,400},
4243 <ptm>      \textquoteleft = {800,500}, \textquoteright = {800,500},
4244 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200},
4245 <cmr>      \textquotedblleft = {700,100}, \textquotedblright = {500,300},
4246 <pad>      \textquotedblleft = {700,200}, \textquotedblright = {700,200},
4247 <ppl>      \textquotedblleft = {500,300}, \textquotedblright = {500,300},
4248 <ptm>      \textquotedblleft = {700,400}, \textquotedblright = {700,400},
4249 <ugm>      \textquotedblleft = {600,200}, \textquotedblright = {600,200},
4250 }
4251
4252 <*cmr>
4253 \SetProtrusion
4254 [ name = cmr-it-OT1,
4255   load = cmr-it ]
4256 { encoding = {OT1,OT4},
4257   family = cmr,
4258   shape = it }
4259 {
4260   \AE = {100, },
4261   \OE = {100, },
4262   "00 = {200,150}, % \Gamma
4263   "01 = {150,100}, % \Delta
4264   "02 = {150, 50}, % \Theta
4265   "03 = {150, 50}, % \Lambda

```

```

4266 "04 = {100,100}, % \Xi
4267 "05 = {100,100}, % \Pi
4268 "06 = {100, 50}, % \Sigma
4269 "07 = {200,150}, % \Upsilon
4270 "08 = {150, 50}, % \Phi
4271 "09 = {150,100}, % \Psi
4272 "0A = { 50, 50}, % \Omega
4273 }
4274
4275 </cmr>
4276 \SetProtrusion
4277 <m-t> [ name = Tl-it-default,
4278 <bch> [ name = bch-it-Tl,
4279 <cmr> [ name = cmr-it-Tl,
4280 <pad> [ name = pad-it-Tl,
4281 <pmn> [ name = pmnj-it-Tl,
4282 <ppl> [ name = ppl-it-Tl,
4283 <ptm> [ name = ptm-it-Tl,
4284 <ugm> [ name = ugm-it-Tl,
4285 <m-t> load = OTl-it ]
4286 <bch> load = bch-it ]
4287 <cmr> load = cmr-it ]
4288 <pmn> load = pmnj-it ]
4289 <pad> load = pad-it ]
4290 <ppl> load = ppl-it ]
4291 <ptm> load = ptm-it ]
4292 <ugm> load = ugm-it ]
4293 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {Tl,Ly1},
4294 <ptm|ugm> { encoding = {Tl},
4295 <bch> family = bch,
4296 <cmr> family = cmr,
4297 <pmn> family = pmnj,
4298 <pad> family = {pad,padx,padj},
4299 <ppl> family = {ppl,pplx,pplj},
4300 <ptm> family = {ptm,ptmx,ptmj},
4301 <ugm> family = ugm,
4302 <!(cmr|ugm)> shape = {it,sl} }
4303 <cmr|ugm> shape = it }
4304 {
4305 <m-t|bch|pmn> _ = { ,100},
4306 <cmr|ugm> _ = {100,200},
4307 <pad|ppl|ptm> _ = {100,100},
4308 <cmr> \AE = {100, },
4309 <bch> \OE = { 50, },
4310 <cmr> \OE = {100, },
4311 <pmn> 031 = { , -100}, % ffi
4312 <cmr|ptm> 156 = {100, }, % IJ
4313 <pad> 156 = {50, }, % IJ
4314 <pmn> 156 = {20, }, % IJ
4315 <pmn> 188 = { , -30}, % ij
4316 <pmn> \v t = { ,100},
4317 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
4318 <cmr|ugm> \textbackslash = {300,300},
4319 <bch> \textbackslash = {150,150},
4320 <pmn> \textbackslash = {100,150},
4321 <ugm> \textbar = {200,200},
4322 <cmr> \textquotedblleft = {500,300},
4323 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
4324 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4325 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4326 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},

```

```

4327 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
4328 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4329 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4330 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4331 <pad> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
4332 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4333 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
4334 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
4335 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4336 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
4337 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
4338 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4339 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
4340 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4341 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
4342 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
4343 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
4344 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
4345 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
4346 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100},
4347 <pmn> \textvisiblespace = {100,100},
4348 }
4349
4350 <*m-t|ptm>
4351 \SetProtrusion
4352 <m-t> [ name = QX-it-default,
4353 <ptm> [ name = ptm-it-QX,
4354 <m-t> load = OT1-it ]
4355 <ptm> load = ptm-it ]
4356 { encoding = {QX},
4357 <ptm> family = {ptm,ptmx,ptmj},
4358 shape = {it,sl} }
4359 {
4360 <ptm> 009 = { , 50}, % fk
4361 {=} = {100,100},
4362 <m-t> \textunderscore = {100,100},
4363 <ptm> \textunderscore = {100,150},
4364 \textbackslash = {100,200},
4365 \quotedblbase = {300,400},
4366 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
4367 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
4368 \textexclamdown = {200, }, \textquestiondown = {200, },
4369 \textbraceleft = {200,100}, \textbraceright = {200,200},
4370 \textless = {100,100}, \textgreater = {100,100},
4371 \textminus = {200,200}, \textdegree = {300,150},
4372 <m-t> \copyright = {100,100}, \textregistered = {100,100},
4373 <ptm> \textregistered = {100,150}, \copyright = {100,150},
4374 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
4375 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
4376 <ptm> \texteuro = {200, }, \textellipsis = {100,150},
4377 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
4378 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
4379 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
4380 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
4381 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
4382 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
4383 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
4384 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
4385 <ptm> \textperthousand = { ,50},
4386 }
4387

```

```

4388 </m-t|ptm>
4389 <*cmr|bch>
4390 \SetProtrusion
4391 <cmr> [ name = cmr-it-T5,
4392 <cmr>   load = cmr-it ]
4393 <bch> [ name = bch-it-T5,
4394 <bch>   load = bch-it ]
4395 { encoding = T5,
4396 <bch>   family = bch,
4397 <cmr>   family = cmr,
4398   shape = it }
4399 {
4400 <bch>   _ = { ,100},
4401 <cmr>   _ = {100,200},
4402 <bch>   \textbackslash = {150,150},
4403 <cmr>   \textbackslash = {300,300},
4404 <bch>   \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4405 <cmr>   \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4406 <bch>   \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4407 <cmr>   \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4408 <bch>   \guillemotleft = {200,300}, \guillemotright = {150,400},
4409 <cmr>   \guillemotleft = {400,100}, \guillemotright = {200,300},
4410 <bch>   \textbraceleft = {200, }, \textbraceright = { ,200},
4411 <cmr>   \textbraceleft = {400,100}, \textbraceright = {200,200},
4412 <bch>   \textless = {100, }, \textgreater = { ,100},
4413 <cmr>   \textless = {300,100}, \textgreater = {200,100},
4414 }
4415
4416 </cmr|bch>

```

Slanted is very similar to italic.

```

4417 <*cmr>
4418 \SetProtrusion
4419 [ name = cmr-sl,
4420   load = cmr-it-OT1 ]
4421 { encoding = {OT1,OT4},
4422   family = cmr,
4423   shape = sl }
4424 {
4425   L = { ,50},
4426   f = { , -50},
4427   - = {300, },
4428   \textendash = {400, }, \textemdash = {300, },
4429 }
4430
4431 \SetProtrusion
4432 [ name = cmr-sl-T1,
4433   load = cmr-it-T1 ]
4434 { encoding = {T1,LV1},
4435   family = cmr,
4436   shape = sl }
4437 {
4438   L = { ,50},
4439   f = { , -50},
4440   - = {300, },
4441   \textendash = {400, }, \textemdash = {300, },
4442 }
4443
4444 \SetProtrusion
4445 [ name = cmr-sl-T5,
4446   load = cmr-it-T5 ]

```

```

4447 { encoding = T5,
4448   family   = cmr,
4449   shape    = sl }
4450 {
4451   L = { ,50},
4452   f = { ,-50},
4453   - = {300, },
4454   \textendash = {400, }, \textemdash = {300, },
4455 }
4456
4457 \SetProtrusion
4458 [ name = lmr-it-T1,
4459   load = cmr-it-T1 ]
4460 { encoding = {T1,LV1},
4461   family   = lmr,
4462   shape    = {it,sl} }
4463 {
4464   \textquotedblleft = {700,100},
4465   \quotedblbase      = {600,300},
4466 }
4467

```

Oldstyle numerals are slightly different.

```

4468 \SetProtrusion
4469 [ name = cmr(oldstyle)-it,
4470   load = cmr-it-T1 ]
4471 { encoding = T1,
4472   family   = {hfor,cmor},
4473   shape    = {it,sl} }
4474 {
4475   1 = {250, 50},
4476   2 = {150,-100},
4477   3 = {100,-50},
4478   4 = {150,150},
4479   6 = {200,  },
4480   7 = {200, 50},
4481   8 = {150,-50},
4482   9 = {100, 50},
4483 }
4484
4485 /cmr
4486 *pmn
4487 \SetProtrusion
4488 [ name = pmnx-it,
4489   load = pmnj-it ]
4490 { encoding = OT1,
4491   family   = pmnx,
4492   shape    = {it,sl} }
4493 {
4494   1 = {100,150},
4495 }
4496
4497 \SetProtrusion
4498 [ name = pmnx-it-T1,
4499   load = pmnj-it-T1 ]
4500 { encoding = {T1,LV1},
4501   family   = pmnx,
4502   shape    = {it,sl} }
4503 {
4504   1 = {100,150},
4505 }

```

```

4506
4507 </pmn>
4508 <*ptm>
4509 \SetProtrusion
4510 [ name      = ptm-it-LY1,
4511   load      = ptm-it-T1 ]
4512 { encoding = {LY1},
4513   family   = {ptm,ptmx,ptmj},
4514   shape     = {it,sl} }
4515 {
4516   -                      = {100,100},
4517   \texttrademark         = {100,100},
4518   \textregistered        = {100,100},
4519   \textcopyright         = {100,100},
4520   \textdegree            = {300,100},
4521   \textminus             = {200,200},
4522   \textellipsis          = {100,150},
4523   \texteuro              = { , },
4524   \textcent              = {100,100},
4525   \textquotesingle       = {500, },
4526   \textflorin            = {100, 70},
4527   \textdagger            = {150,150},
4528   \textdaggerdbl         = {100,100},
4529   \textbullet            = {150,150},
4530   \textonesuperior       = {150,100},
4531   \texttwosuperior       = {150, 50},
4532   \textthreesuperior     = {150, 50},
4533   \textparagraph         = {100, },
4534   \textperiodcentered    = {500,300},
4535   \textonequarter        = { 50, },
4536   \textonehalf           = { 50, },
4537   \textplusminus         = {100,100},
4538   \textmultiply          = {150,150},
4539   \textdivide            = {150,150},
4540 }
4541
4542 </ptm>

```

### 14.7.3 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).

```

4543 <*!ugm>
4544 \SetProtrusion
4545 <m-t> [ name      = OT1-sc,
4546 <bch> [ name      = bch-sc,
4547 <cmr> [ name      = cmr-sc-OT1,
4548 <pad> [ name      = pad-sc,
4549 <pmn> [ name      = pmnj-sc,
4550 <ppl> [ name      = ppl-sc,
4551 <ptm> [ name      = ptm-sc,
4552 <m-t> load      = default ]
4553 <bch> load      = bch-default ]
4554 <cmr> load      = cmr-OT1 ]
4555 <pad> load      = pad-default ]
4556 <pmn> load      = pmnj-default ]
4557 <ppl> load      = ppl-default ]
4558 <ptm> load      = ptm-default ]

```

```

4559 <m-t|bch|pad|pmn> { encoding = OT1,
4560 <cmr|ppl|ptm> { encoding = {OT1,OT4},
4561 <bch> family = bch,
4562 <cmr> family = cmr,
4563 <pad> family = {pad,padx,padj},
4564 <pmn> family = pmnj,
4565 <ppl> family = {ppl,pplx,pplj},
4566 <ptm> family = {ptm,ptmx,ptmj},
4567 shape = sc }
4568 {
4569 a = {50,50},
4570 <cmr|pad|ppl|ptm> \ae = {50, },
4571 <bch|pmn> c = {50, },
4572 <bch|pad|pmn> d = { ,50},
4573 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
4574 <bch|pad|pmn> g = {50, },
4575 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
4576 <bch> j = {100, },
4577 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
4578 <ptm> l = { ,80},
4579 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
4580 <ptm> 013 = { ,80}, % fl
4581 <bch|pad|pmn> o = {50,50},
4582 <pad|pmn> \oe = {50, },
4583 <ppl> p = { 0, 0},
4584 <bch|pad|pmn> q = {50,70},
4585 <ppl> q = { 0, },
4586 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
4587 t = {50,50},
4588 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50},
4589 <ptm> y = {80,80},
4590 }
4591
4592 \SetProtrusion
4593 <m-t> [ name = T1-sc,
4594 <bch> [ name = bch-sc-T1,
4595 <cmr> [ name = cmr-sc-T1,
4596 <pad> [ name = pad-sc-T1,
4597 <pmn> [ name = pmnj-sc-T1,
4598 <ppl> [ name = ppl-sc-T1,
4599 <ptm> [ name = ptm-sc-T1,
4600 <m-t> load = T1-default ]
4601 <bch> load = bch-T1 ]
4602 <cmr> load = cmr-T1 ]
4603 <pad> load = pad-T1 ]
4604 <pmn> load = pmnj-T1 ]
4605 <ppl> load = ppl-T1 ]
4606 <ptm> load = ptm-T1 ]
4607 { encoding = {T1,LY1},
4608 <bch> family = bch,
4609 <cmr> family = cmr,
4610 <pad> family = {pad,padx,padj},
4611 <pmn> family = pmnj,
4612 <ppl> family = {ppl,pplx,pplj},
4613 <ptm> family = {ptm,ptmx,ptmj},
4614 shape = sc }
4615 {
4616 a = {50,50},
4617 <cmr|pad|ppl|ptm> \ae = {50, },
4618 <bch|pmn> c = {50, },
4619 <bch|pad|pmn> d = { ,50},

```

```

4620 <m-t|bch|cmr|pad|pmn|ptm>    f = { ,50},
4621 <bch|pad|pmn>                g = {50, },
4622 <m-t|cmr|pad|pmn|ppl|ptm>    j = {50, },
4623 <bch>                        j = {100, },
4624 <m-t|bch|cmr|pad|pmn|ppl>    l = { ,50},
4625 <ptm>                        l = { ,80},
4626 <m-t|bch|cmr|pad|pmn|ppl>    029 = { ,50}, % f1
4627 <ptm>                        029 = { ,80}, % f1
4628 <bch|pad|pmn>                o = {50,50},
4629 <bch|pad|pmn>                \oe = {50, },
4630 <ppl>                        p = { 0, 0},
4631 <bch|pad|pmn>                q = {50,70},
4632 <ppl>                        q = { 0, },
4633 <m-t|cmr|pad|pmn|ppl|ptm>    r = { , 0},
4634 t = {50,50},
4635 <m-t|bch|cmr|pad|pmn|ppl>    y = {50,50},
4636 <ptm>                        y = {80,80},
4637 }
4638
4639 </!ugm>
4640 <*m-t>
4641 \SetProtrusion
4642 [ name      = QX-sc,
4643   load      = QX-default ]
4644 { encoding = QX,
4645   shape    = sc }
4646 {
4647   a = {50,50},
4648   f = { ,50},
4649   j = {50, },
4650   l = { ,50},
4651   013 = { ,50}, % f1
4652   r = { , 0},
4653   t = {50,50},
4654   y = {50,50},
4655 }
4656
4657 </m-t>
4658 <*cmr|bch>
4659 \SetProtrusion
4660 <bch> [ name      = bch-sc-T5,
4661 <bch>   load      = bch-T5 ]
4662 <cmr> [ name      = cmr-sc-T5,
4663 <cmr>   load      = cmr-T5 ]
4664 { encoding = T5,
4665 <bch>   family   = bch,
4666 <cmr>   family   = cmr,
4667   shape    = sc }
4668 {
4669   a = {50,50},
4670 <bch>   c = {50, },
4671 <bch>   d = { ,50},
4672   f = { ,50},
4673 <bch>   g = {50, },
4674 <bch>   j = {100, },
4675 <cmr>   j = {50, },
4676   l = { ,50},
4677 <bch>   o = {50,50},
4678 <bch>   q = { 0, },
4679 <cmr>   r = { , 0},
4680   t = {50,50},

```



```

4681     y = {50,50},
4682   }
4683
4684 /cmr|bch
4685 *pmn
4686 \SetProtrusion
4687   [ name      = pmnx-sc,
4688     load      = pmnj-sc ]
4689   { encoding = OT1,
4690     family   = pmnx,
4691     shape    = sc }
4692   {
4693     1 = {230,180},
4694   }
4695
4696 \SetProtrusion
4697   [ name      = pmnx-sc-T1,
4698     load      = pmnj-sc-T1 ]
4699   { encoding = {T1,LV1},
4700     family   = pmnx,
4701     shape    = sc }
4702   {
4703     1 = {230,180},
4704   }
4705

```

#### 14.7.4 Italic Small Caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's `fontinstallationguide` suggests `si`.

```

4706 \SetProtrusion
4707   [ name      = pmnj-scit,
4708     load      = pmnj-it ]
4709   { encoding = OT1,
4710     family   = pmnj,
4711     shape    = {scit,si} }
4712   {
4713     a = {50, },
4714     \ae = { , -50},
4715     b = {20,-50},
4716     c = {50,-50},
4717     d = {20, 0},
4718     e = {20,-50},
4719     f = {10, 0},
4720     012 = {10,-50}, % fi
4721     013 = {10,-50}, % fl
4722     014 = {10,-50}, % ffi
4723     015 = {10,-50}, % ffl
4724     g = {50,-50},
4725     i = {20,-50},
4726     j = {20, 0},
4727     k = {20, },
4728     l = {20,50},
4729     m = { , -30},
4730     n = { , -30},
4731     o = {50, },
4732     \oe = {50,-50},
4733     p = {20,-50},
4734     q = {50, },
4735     r = {20, 0},

```

```

4736     s = {20,-30},
4737     t = {70,  },
4738     u = {50,-50},
4739     v = {100,  },
4740     w = {100,  },
4741     y = {50,  },
4742     z = {  ,-50},
4743 }
4744
4745 \SetProtrusion
4746 [ name      = pmnj-scit-T1,
4747   load      = pmnj-it-T1 ]
4748 { encoding = {T1,LY1},
4749   family   = pmnj,
4750   shape     = {scit,si} }
4751 {
4752   a = {50,  },
4753   \ae = {  ,-50},
4754   b = {20,-50},
4755   c = {50,-50},
4756   d = {20, 0},
4757   e = {20,-50},
4758   f = {10, 0},
4759   028 = {10,-50}, % fi
4760   029 = {10,-50}, % fl
4761   030 = {10,-50}, % ffi
4762   031 = {10,-50}, % ffl
4763   g = {50,-50},
4764   i = {20,-50},
4765   188 = {20, 0}, % ij
4766   j = {20, 0},
4767   k = {20,  },
4768   l = {20,50},
4769   m = {  ,-30},
4770   n = {  ,-30},
4771   o = {50,  },
4772   \oe = {50,-50},
4773   p = {20,-50},
4774   q = {50,  },
4775   r = {20, 0},
4776   s = {20,-30},
4777   t = {70,  },
4778   u = {50,-50},
4779   v = {100,  },
4780   w = {100,  },
4781   y = {50,  },
4782   z = {  ,-50},
4783 }
4784
4785 \SetProtrusion
4786 [ name      = pmnx-scit,
4787   load      = pmnj-scit ]
4788 { encoding = OT1,
4789   family   = pmnx,
4790   shape     = {scit,si} }
4791 {
4792   l = {100,150},
4793 }
4794
4795 \SetProtrusion
4796 [ name      = pmnx-scit-T1,

```

```

4797     load      = pmnj-scit-T1 ]
4798     { encoding = {T1,LY1},
4799       family   = pmnx,
4800       shape    = {scit,si}    }
4801     {
4802       1 = {100,150},
4803     }
4804
4805 </pmn>

```

### 14.7.5 textcomp

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

4806 \SetProtrusion
4807 <m-t> [ name      = textcomp ]
4808 <bch> [ name      = bch-textcomp ]
4809 <cmr> [ name      = cmr-textcomp ]
4810 <pad> [ name      = pad-textcomp ]
4811 <pmn> [ name      = pmn-textcomp ]
4812 <ppl> [ name      = ppl-textcomp ]
4813 <ptm> [ name      = ptm-textcomp ]
4814 <ugm> [ name      = ugm-textcomp ]
4815 <m-t> { encoding = TS1      }
4816 <!m-t> { encoding = TS1,
4817 <bch>     family   = bch }
4818 <cmr>     family   = cmr }
4819 <pad>     family   = {pad,padx,padj} }
4820 <pmn>     family   = {pmnx,pmnj} }
4821 <ppl>     family   = {ppl,pplx,pplj} }
4822 <ptm>     family   = {ptm,ptmx,ptmj} }
4823 <ugm>     family   = ugm }
4824 {
4825 <cmr>     \textquotestraightbase = {300,300},
4826 <pad|pmn> \textquotestraightbase = {400,400},
4827 <cmr|pmn> \textquotestraightdblbase = {300,300},
4828 <pad>     \textquotestraightdblbase = {400,400},
4829 <bch|cmr|pad|pmn|ugm> \texttwelveudash = {200,200},
4830 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
4831 <ugm>     \textthreequartersemdash = {200,200},
4832 <cmr|pmn> \textquotesingle = {300,400},
4833 <pad>     \textquotesingle = {400,500},
4834 <ptm>     \textquotesingle = {500,500},
4835 <ugm>     \textquotesingle = {300,500},
4836 <bch|cmr|pmn> \textasteriskcentered = {200,300},
4837 <pad>     \textasteriskcentered = {300,300},
4838 <ugm>     \textasteriskcentered = {100,200},
4839 <pmn>     \textfractionsolidus = {-200,-200},
4840 <cmr>     \textoneoldstyle = {100,100},
4841 <pmn>     \textoneoldstyle = { , 50},
4842 <cmr>     \textthreeoldstyle = { , 50},
4843 <pad|pmn> \textthreeoldstyle = { 50, },
4844 <cmr>     \textfouroldstyle = { 50, 50},
4845 <pad|pmn> \textfouroldstyle = { 50, },
4846 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
4847 <cmr>     \textlangle = {400, },
4848 <cmr>     \textrightangle = { ,400},
4849 <m-t|bch|pmn|ptm> \textminus = {200,200},
4850 <cmr|pad|ppl> \textminus = {300,300},
4851 <ugm>     \textminus = {250,300},

```

```

4852 <bch|pad|pmn> \textlbrackdbl = {100, },
4853 <bch|pad|pmn> \textrbrackdbl = { ,100},
4854 <pmn> \textasciigrave = {200,500},
4855 <bch|cmr|pad|pmn> \texttildebelow = {200,250},
4856 <pmn> \textasciibreve = {300,400},
4857 <pmn> \textasciicaron = {300,400},
4858 <pmn> \textacutedbl = {200,300},
4859 <pmn> \textgravedbl = {150,300},
4860 <bch|pmn|ugm> \textdagger = { 80, 80},
4861 <cmr|pad> \textdagger = {100,100},
4862 <ptm> \textdagger = {150,150},
4863 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
4864 <ptm> \textdaggerdbl = {100,100},
4865 <bch> \textbardbl = {100,100},
4866 <ugm> \textbardbl = {150,150},
4867 <bch> \textbullet = {200,200},
4868 <cmr|pad|pmn> \textbullet = { ,100},
4869 <ptm> \textbullet = {150,150},
4870 <ugm> \textbullet = { 50,100},
4871 <bch|cmr|pmn> \textcelsius = { 50, },
4872 <pad> \textcelsius = { 80, },
4873 <bch> \textflorin = { 50, 50},
4874 <pad|ugm> \textflorin = { ,100},
4875 <pmn> \textflorin = { 50,100},
4876 <ptm> \textflorin = { 50, 70},
4877 <cmr> \textcolonmonetary = { , 50},
4878 <pad|pmn> \textcolonmonetary = { 50, },
4879 <pmn> \textinterrobang = { ,100},
4880 <pmn> \textinterrobangdown = {100, },
4881 <m-t|pad|ptm> \texttrademark = {100,100},
4882 <bch> \texttrademark = {150,150},
4883 <cmr|ppl> \texttrademark = {200,200},
4884 <pmn> \texttrademark = { 50, 50},
4885 <ugm> \texttrademark = {100,150},
4886 <bch|ugm> \textcent = { 50, },
4887 <ptm> \textcent = {100,100},
4888 <bch> \textsterling = { 50, },
4889 <ugm> \textsterling = { , 50},
4890 <bch> \textbrokenbar = {200,200},
4891 <ugm> \textbrokenbar = {200,300},
4892 <pmn> \textasciidieresis = {300,400},
4893 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
4894 <pmn> \textcopyright = {100,150},
4895 <ppl> \textcopyright = {200,200},
4896 <bch|cmr|ugm> \textordfeminine = {100,200},
4897 <pad|pmn> \textordfeminine = {200,200},
4898 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
4899 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
4900 <pmn> \textregistered = { 50,150},
4901 <ppl> \textregistered = { 200,200},
4902 <pmn> \textasciimacron = {150,200},
4903 <m-t|ppl|ptm> \textdegree = {300,300},
4904 <bch> \textdegree = {150,200},
4905 <cmr|pad> \textdegree = {400,400},
4906 <pmn> \textdegree = {150,400},
4907 <ugm> \textdegree = {200,200},
4908 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
4909 <ptm> \textpm = { 50, 80},
4910 <bch|ugm> \texttwosuperior = {100,200},
4911 <cmr> \texttwosuperior = { 50,100},
4912 <pad|pmn> \texttwosuperior = {200,200},

```

```

4913 <ptm> \texttwosuperior = { 50, 50},
4914 <bch|ugm> \textthreesuperior = {100,200},
4915 <cmr> \textthreesuperior = { 50,100},
4916 <pad|pmn> \textthreesuperior = {200,200},
4917 <ptm> \textthreesuperior = { 50, 50},
4918 <pmn> \textasciicute = {300,400},
4919 <bch|ugm> \textmu = { ,100},
4920 <bch|pad|pmn> \textparagraph = { ,100},
4921 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
4922 <ptm> \textperiodcentered = {300,300},
4923 <ugm> \textperiodcentered = {200,500},
4924 <bch|ugm> \textonesuperior = {200,300},
4925 <cmr|pad|pmn> \textonesuperior = {200,200},
4926 <ptm> \textonesuperior = {100,100},
4927 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
4928 <cmr> \textordmasculine = {100,200},
4929 <bch|cmr|pmn> \texteuro = {100, },
4930 <pad> \texteuro = { 50,100},
4931 <bch|ptm> \texttimes = {100,100},
4932 <cmr> \texttimes = {150,250},
4933 <pad> \texttimes = {100,150},
4934 <pmn> \texttimes = { 70,100},
4935 <ugm> \texttimes = {200,300},
4936 <bch|pad|pmn> \textdiv = {150,200},
4937 <cmr> \textdiv = {150,250},
4938 <ptm> \textdiv = { 50,100},
4939 <ugm> \textdiv = {200,300},
4940 <ptm> \textperthousand = { ,50},
4941 <ugm> \textsection = { ,100},
4942 <ugm> \textonehalf = { 50,100},
4943 <ugm> \textonequarter = { 50,100},
4944 <ugm> \textthreequarters = { 50,100},
4945 <ugm> \textsurd = { ,100},

```

All remaining characters can be found in the source.

```

4946 }
4947
4948 <*cmr|pad|pmn|ugm>
4949 \SetProtrusion
4950 <cmr> [ name = cmr-textcomp-it ]
4951 <pad> [ name = pad-textcomp-it ]
4952 <pmn> [ name = pmn-textcomp-it ]
4953 <ugm> [ name = ugm-textcomp-it ]
4954 { encoding = TS1,
4955 <cmr> family = cmr,
4956 <pad> family = {pad,padx,padj},
4957 <pmn> family = {pmnx,pmnj},
4958 <ugm> family = ugm,
4959 <!ugm> shape = {it,sl} }
4960 <ugm> shape = it }
4961 {
4962 <cmr> \textquotestraightbase = {300,600},
4963 <pad|pmn> \textquotestraightbase = {400,400},
4964 <cmr> \textquotestraightdblbase = {300,600},
4965 <pad> \textquotestraightdblbase = {300,400},
4966 <pmn> \textquotestraightdblbase = {300,300},
4967 \texttwelveudash = {200,200},
4968 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
4969 <ugm> \textthreequartersemdash = {200,200},
4970 <cmr> \textquotesingle = {600,300},
4971 <pad> \textquotesingle = {800,100},

```

```

4972 <pmn> \textquotesingle = {300,200},
4973 <ugm> \textquotesingle = {500,500},
4974 <cmr> \textasteriskcentered = {300,200},
4975 <pad> \textasteriskcentered = {500,100},
4976 <pmn> \textasteriskcentered = {200,300},
4977 <ugm> \textasteriskcentered = {300,150},
4978 <pmn> \textfractionsolidus = {-200,-200},
4979 <cmr> \textoneoldstyle = {100, 50},
4980 <pad> \textoneoldstyle = {100, },
4981 <pmn> \textoneoldstyle = { 50, },
4982 <pad> \texttwooldstyle = { 50, },
4983 <pmn> \texttwooldstyle = {-50, },
4984 <cmr> \textthreeoldstyle = {100, 50},
4985 <pmn> \textthreeoldstyle = {-100, },
4986 <cmr> \textfouroldstyle = { 50, 50},
4987 <pad> \textfouroldstyle = { 50,100},
4988 <cmr> \textsevenoldstyle = { 50, 80},
4989 <pad> \textsevenoldstyle = { 50, },
4990 <pmn> \textsevenoldstyle = { 20, },
4991 <cmr> \textlangle = {400, },
4992 <cmr> \textrangle = { ,400},
4993 <cmr|pad> \textminus = {300,300},
4994 <pmn> \textminus = {200,200},
4995 <ugm> \textminus = {250,300},
4996 <pad|pmn> \textlbrackdbl = {100, },
4997 <pad|pmn> \textrbrackdbl = { ,100},
4998 <pmn> \textasciigrave = {300,300},
4999 <cmr|pad|pmn> \texttildelow = {200,250},
5000 <pmn> \textasciibreve = {300,300},
5001 <pmn> \textasciicaron = {300,300},
5002 <pmn> \textacutedbl = {200,300},
5003 <pmn> \textgravedbl = {150,300},
5004 <cmr> \textdagger = {100,100},
5005 <pad> \textdagger = {200,100},
5006 <pmn> \textdagger = { 80, 50},
5007 <ugm> \textdagger = { 80, 80},
5008 <cmr|pad> \textdaggerdbl = { 80, 80},
5009 <pmn> \textdaggerdbl = { 80, 50},
5010 <ugm> \textbardbl = {150,150},
5011 <cmr> \textbullet = {200,100},
5012 <pad> \textbullet = {300, },
5013 <pmn> \textbullet = { 30, 70},
5014 <ugm> \textbullet = { 50,100},
5015 <cmr> \textcelsius = {100, },
5016 <pad> \textcelsius = {200, },
5017 <pmn> \textcelsius = { 50,-50},
5018 <pad> \textflorin = {100, },
5019 <pmn> \textflorin = { 50,100},
5020 <ugm> \textflorin = { ,100},
5021 <cmr> \textcolonmonetary = {150, },
5022 <pad> \textcolonmonetary = {100, },
5023 <pmn> \textcolonmonetary = { 50,-50},
5024 <cmr|pad> \texttrademark = {200, },
5025 <pmn> \texttrademark = { 50,100},
5026 <ugm> \texttrademark = {150, 50},
5027 <ugm> \textcent = { 50, },
5028 <ugm> \textsterling = { , 50},
5029 <ugm> \textbrokenbar = {200,300},
5030 <pmn> \textasciidieresis = {300,200},
5031 <cmr> \textcopyright = {100, },
5032 <pad> \textcopyright = {200,100},

```

```

5033 <pmn> \textcopyright = {100,150},
5034 <ugm> \textcopyright = {300, },
5035 <cmr> \textordfeminine = {100,100},
5036 <pmn> \textordfeminine = {200,200},
5037 <ugm> \textordfeminine = {100,200},
5038 <cmr|pad> \textlnot = {300, },
5039 <pmn|ugm> \textlnot = {200, },
5040 <cmr> \textregistered = {100, },
5041 <pad> \textregistered = {200,100},
5042 <pmn> \textregistered = { 50,150},
5043 <ugm> \textregistered = {300, },
5044 <pmn> \textasciimacron = {150,200},
5045 <cmr|pad> \textdegree = {500,100},
5046 <pmn> \textdegree = {150,150},
5047 <ugm> \textdegree = {300,200},
5048 <cmr> \textpm = {150,100},
5049 <pad> \textpm = {200,150},
5050 <pmn|ugm> \textpm = {150,200},
5051 <cmr> \textonesuperior = {400, },
5052 <pad> \textonesuperior = {300,100},
5053 <pmn> \textonesuperior = {200,100},
5054 <ugm> \textonesuperior = {300,300},
5055 <cmr> \texttwosuperior = {400, },
5056 <pad> \texttwosuperior = {300, },
5057 <pmn> \texttwosuperior = {200,100},
5058 <ugm> \texttwosuperior = {300,200},
5059 <cmr> \textthreesuperior = {400, },
5060 <pad> \textthreesuperior = {300, },
5061 <pmn> \textthreesuperior = {200,100},
5062 <ugm> \textthreesuperior = {300,200},
5063 <ugm> \textmu = { ,100},
5064 <pmn> \textasciicute = {300,200},
5065 <cmr> \textparagraph = {200, },
5066 <pmn> \textparagraph = { ,100},
5067 <cmr> \textperiodcentered = {500,500},
5068 <pad|pmn|ugm> \textperiodcentered = {300,400},
5069 <cmr> \textordmasculine = {100,100},
5070 <pmn> \textordmasculine = {200,200},
5071 <ugm> \textordmasculine = {300,200},
5072 <cmr> \texteuro = {200, },
5073 <pad> \texteuro = {100, },
5074 <pmn> \texteuro = {100,-50},
5075 <cmr> \texttimes = {200,200},
5076 <pad> \texttimes = {200,100},
5077 <pmn> \texttimes = { 70,100},
5078 <ugm> \texttimes = {200,300},
5079 <cmr|pad> \textdiv = {200,200},
5080 <pmn> \textdiv = {150,200},
5081 <ugm> \textdiv = {200,300},
5082 <ugm> \textsection = { ,200},
5083 <ugm> \textonehalf = { 50,100},
5084 <ugm> \textonequarter = { 50,100},
5085 <ugm> \textthreequarters = { 50,100},
5086 <ugm> \textsurd = { ,100},
5087 }
5088
5089 </cmr|pad|pmn|ugm>

```

### 14.7.6 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}
```

```
5090 <*cmr>
5091 \SetProtrusion
5092 [ name = cmr-math-letters ]
5093 { encoding = OML,
5094   family = cmm,
5095   series = {m,b},
5096   shape = it }
5097 {
5098   A = {100, 50}, % \mathnormal
5099   B = { 50, },
5100   C = { 50, },
5101   D = { 50, 50},
5102   E = { 50, },
5103   F = {100, 50},
5104   G = { 50, 50},
5105   H = { 50, 50},
5106   I = { 50, 50},
5107   J = {150, 50},
5108   K = { 50,100},
5109   L = { 50, 50},
5110   M = { 50, },
5111   N = { 50, },
5112   O = { 50, },
5113   P = { 50, },
5114   Q = { 50, 50},
5115   R = { 50, },
5116   S = { 50, },
5117   T = { 50,100},
5118   U = { 50, 50},
5119   V = {100,100},
5120   W = { 50,100},
5121   X = { 50,100},
5122   Y = {100,100},
5123   f = {100,100},
5124   h = { ,100},
5125   i = { , 50},
5126   j = { , 50},
5127   k = { , 50},
5128   r = { , 50},
5129   v = { , 50},
5130   w = { , 50},
5131   x = { , 50},
```



```

5132 "0B = { 50,100}, % \alpha
5133 "0C = { 50, 50}, % \beta
5134 "0D = {200,150}, % \gamma
5135 "0E = { 50, 50}, % \delta
5136 "0F = { 50, 50}, % \epsilon
5137 "10 = { 50,150}, % \zeta
5138 % "11 = { , }, % \eta
5139 "12 = { 50, }, % \theta
5140 "13 = { ,100}, % \iota
5141 "14 = { ,100}, % \kappa
5142 "15 = {100, 50}, % \lambda
5143 "16 = { , 50}, % \mu
5144 "17 = { , 50}, % \nu
5145 "18 = { , 50}, % \xi
5146 "19 = { 50,100}, % \pi
5147 "1A = { 50, 50}, % \rho
5148 "1B = { ,150}, % \sigma
5149 "1C = { 50,150}, % \tau
5150 "1D = { 50, 50}, % \upsilon
5151 % "1E = { , }, % \phi
5152 "1F = { 50,100}, % \chi
5153 "20 = { 50, 50}, % \psi
5154 "21 = { , 50}, % \omega
5155 "22 = { , 50}, % \varepsilon
5156 "23 = { , 50}, % \vartheta
5157 "24 = { , 50}, % \varpi
5158 "25 = {100, }, % \varrho
5159 "26 = {100,100}, % \varsigma
5160 "27 = { 50, 50}, % \varphi
5161 "28 = {100,100}, % \leftharpoonup
5162 "29 = {100,100}, % \leftharpoondown
5163 "2A = {100,100}, % \rightharpoonup
5164 "2B = {100,100}, % \rightharpoondown
5165 "2C = {300,200}, % \lhook
5166 "2D = {200,300}, % \rhook
5167 "2E = { ,100}, % \triangleright
5168 "2F = {100, }, % \triangleleft
5169 % 0 - 9
5170 "3A = { ,500}, % ., \ldotp
5171 "3B = { ,500}, % ,
5172 "3C = {200,100}, % <
5173 "3D = {300,400}, % /
5174 "3E = {100,200}, % >
5175 "3F = {200,200}, % \star
5176 % "40 = { , }, % \partial
5177 "5B = { ,100}, % \flat
5178 % "5C = { , }, % \natural
5179 % "5D = { , }, % \sharp
5180 "5E = {200,200}, % \smile
5181 "5F = {200,200}, % \frown
5182 % "60 = { , }, % \ell
5183 % "7B = { , }, % \imath
5184 "7C = {100, }, % \jmath
5185 "7D = { ,100}, % \wp
5186 }
5187

```

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}

```

```

5188 \SetProtrusion
5189 [ name      = cmr-math-symbols ]
5190 { encoding = OMS,
5191   family   = cmsy,
5192   series    = {m,b},
5193   shape     = n }
5194 {
5195     A = {150, 50}, % \mathcal
5196     C = {   ,100},
5197     D = {   , 50},
5198     F = { 50,150},
5199     I = {   ,100},
5200     J = {100,150},
5201     K = {   ,100},
5202     L = {100,   },
5203     M = { 50, 50},
5204     N = { 50,100},
5205     P = {   , 50},
5206     Q = { 50,   },
5207     R = {   , 50},
5208     T = { 50,150},
5209     V = { 50, 50},
5210     W = {   , 50},
5211     X = {100,100},
5212     Y = {100,   },
5213     Z = {100,150},
5214     "00 = {300,300}, % -
5215     "01 = {   ,700}, % \cdot, \cdotp
5216     "02 = {150,250}, % \times
5217     "03 = {150,250}, % *, \ast
5218     "04 = {200,300}, % \div
5219     "05 = {150,250}, % \diamond
5220     "06 = {200,200}, % \pm
5221     "07 = {200,200}, % \mp
5222     "08 = {100,100}, % \oplus
5223     "09 = {100,100}, % \ominus
5224     "0A = {100,100}, % \otimes
5225     "0B = {100,100}, % \oslash
5226     "0C = {100,100}, % \odot
5227     "0D = {100,100}, % \bigcirc
5228     "0E = {100,100}, % \circ
5229     "0F = {100,100}, % \bullet
5230     "10 = {100,100}, % \asymp
5231     "11 = {100,100}, % \equiv
5232     "12 = {200,100}, % \subseteq
5233     "13 = {100,200}, % \supseteq
5234     "14 = {200,100}, % \leq
5235     "15 = {100,200}, % \geq
5236     "16 = {200,100}, % \preceq
5237     "17 = {100,200}, % \succeq
5238     "18 = {200,200}, % \sim
5239     "19 = {150,150}, % \approx
5240     "1A = {200,100}, % \subset
5241     "1B = {100,200}, % \supset
5242     "1C = {200,100}, % \ll
5243     "1D = {100,200}, % \gg
5244     "1E = {300,100}, % \prec
5245     "1F = {100,300}, % \succ
5246     "20 = {100,200}, % \leftarrow
5247     "21 = {200,100}, % \rightarrow
5248     "22 = {100,100}, % \uparrow

```

```

5249 "23 = {100,100}, % \downarrow
5250 "24 = {100,100}, % \leftrightarrows
5251 "25 = {100,100}, % \nearrow
5252 "26 = {100,100}, % \searrow
5253 "27 = {100,100}, % \simeq
5254 "28 = {100,100}, % \Leftarrow
5255 "29 = {100,100}, % \Rightarrow
5256 "2A = {100,100}, % \Uparrow
5257 "2B = {100,100}, % \Downarrow
5258 "2C = {100,100}, % \Leftrightarrow
5259 "2D = {100,100}, % \nrightarrow
5260 "2E = {100,100}, % \swarrow
5261 "2F = { ,100}, % \propto
5262 "30 = { ,400}, % \prime
5263 "31 = {100,100}, % \infty
5264 "32 = {150,100}, % \in
5265 "33 = {100,150}, % \ni
5266 "34 = {100,100}, % \triangle, \bigtriangleup
5267 "35 = {100,100}, % \bigtriangledown
5268 % "36 = { , }, % \not
5269 % "37 = { , }, % \mapstochar
5270 "38 = { ,100}, % \forall
5271 "39 = {100, }, % \exists
5272 "3A = {200, }, % \neg
5273 % "3B = { , }, % \emptyset
5274 % "3C = { , }, % \Re
5275 % "3D = { , }, % \Im
5276 "3E = {200,200}, % \top
5277 "3F = {200,200}, % \bot, \perp
5278 % "40 = { , }, % \aleph
5279 % "5B = { , }, % \cup
5280 % "5C = { , }, % \cap
5281 % "5D = { , }, % \uplus
5282 "5E = {100,200}, % \wedge
5283 "5F = {100,200}, % \vee
5284 "60 = { ,300}, % \vdash
5285 "61 = {300, }, % \dashv
5286 "62 = {100,100}, % \lfloor
5287 "63 = {100,100}, % \rfloor
5288 "64 = {100,100}, % \lceil
5289 "65 = {100,100}, % \rceil
5290 "66 = {150, }, % \lbrace
5291 "67 = { ,150}, % \rbrace
5292 "68 = {400, }, % \langle
5293 "69 = { ,400}, % \rangle
5294 % "6A = { , }, % \arrowvert, \mid, \vert, |
5295 % "6B = { , }, % \Arrowvert, \parallel, \Vert
5296 "6C = {100,100}, % \updownarrow
5297 "6D = {100,100}, % \Updownarrow
5298 "6E = {100,300}, % \, \backslash, \setminus
5299 % "6F = { , }, % \wr
5300 % "70 = { , }, % \sqrt{\phantom{x}}
5301 % "71 = { , }, % \amalg
5302 "72 = {100,100}, % \nabla
5303 % "73 = { , }, % \smallint
5304 % "74 = { , }, % \sqcup
5305 % "75 = { , }, % \sqcap
5306 % "76 = { , }, % \sqsubseteq
5307 % "77 = { , }, % \sqsupseteq
5308 % "78 = { , }, % \mathsection
5309 "79 = {200,200}, % \dagger

```

```

5310 "7A = {100,100}, % \ddagger
5311 "7B = {100, }, % \mathparagraph
5312 "7C = {100,100}, % \clubsuit
5313 "7D = {100,100}, % \diamondsuit
5314 "7E = {100,100}, % \heartsuit
5315 "7F = {100,100}, % \spadesuit
5316 }
5317

```

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}
```

```

5318 </cmr>
5319 </cfg-t>

```

### 14.7.7 AMS fonts

Settings for the AMS math fonts.

```
5320 <*cfg-u>
```

Symbol font 'a', defined in amssymb.

```
\DeclareSymbolFont{AMSa}{U}{msa}{m}{n}
```

```

5321 <*msa>
5322 \SetProtrusion
5323 [ name = AMS-a ]
5324 { encoding = U,
5325   family = msa }
5326 {
5327 % "00 = { , }, % \boxdot
5328 % "01 = { , }, % \boxplus
5329 % "02 = { , }, % \boxtimes
5330 % "03 = { , }, % \square
5331 % "04 = { , }, % \blacksquare
5332 % "05 = {150,250}, % \centerdot
5333 % "06 = {100,100}, % \lozenge
5334 % "07 = { 50, 50}, % \blacklozenge
5335 % "08 = { 50, 50}, % \circlearrowright
5336 % "09 = { 50, 50}, % \circlearrowleft
5337 % "0A = {100,100}, % \rightleftharpoons
5338 % "0B = {100,100}, % \leftrightharpoons
5339 % "0C = { , }, % \boxminus
5340 % "0D = {-50,200}, % \Vdash
5341 % "0E = {-50,200}, % \Vvdash
5342 % "0F = {-70,150}, % \vdash
5343 % "10 = {100,150}, % \twoheadrightarrow
5344 % "11 = {100,150}, % \twoheadleftarrow
5345 % "12 = { 50,100}, % \leftleftarrows
5346 % "13 = { 50, 80}, % \rightrightarrows
5347 % "14 = {120,120}, % \upuparrows
5348 % "15 = {120,120}, % \downdownarrows
5349 % "16 = {200,200}, % \upharpoonright
5350 % "17 = {200,200}, % \downharpoonright
5351 % "18 = {200,200}, % \upharpoonleft
5352 % "19 = {200,200}, % \downharpoonleft
5353 % "1A = { 80,100}, % \rightarrowtail
5354 % "1B = { 80,100}, % \leftarrowtail
5355 % "1C = { 50, 50}, % \leftrightarrows

```

```

5356 "1D = { 50, 50}, % \rightleftarrows
5357 "1E = {250, }, % \Lsh
5358 "1F = { ,250}, % \Rsh
5359 "20 = {100,100}, % \rightsquigarrow
5360 "21 = {100,100}, % \leftrightsquigarrow
5361 "22 = {100, 50}, % \looparrowleft
5362 "23 = { 50,100}, % \looparrowright
5363 "24 = { 50, 80}, % \circeq
5364 "25 = { ,100}, % \succsim
5365 "26 = { ,100}, % \gtrsim
5366 "27 = { ,100}, % \gtrapprox
5367 "28 = {150, 50}, % \multimap
5368 % "29 = { , }, % \therefore
5369 % "2A = { , }, % \because
5370 "2B = {100,150}, % \doteqdot
5371 "2C = {100,150}, % \triangleq
5372 "2D = {100, 50}, % \precsim
5373 "2E = {100, 50}, % \lessim
5374 "2F = { 50, 50}, % \lessapprox
5375 "30 = {100, 50}, % \eqslantless
5376 "31 = { 50, 50}, % \eqslantgtr
5377 "32 = {100, 50}, % \curlyeqprec
5378 "33 = { 50,100}, % \curlyeqsucc
5379 "34 = {100, 50}, % \preccurlyeq
5380 % "35 = { , }, % \leqq
5381 "36 = { 50, }, % \leqslant
5382 % "37 = { , }, % \lessgtr
5383 "38 = { , 50}, % \backprime
5384 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
5385 % "3A = { , }, % \risingdotseq
5386 % "3B = { , }, % \fallingdotseq
5387 "3C = { 50,100}, % \succcurlyeq
5388 % "3D = { , }, % \geqq
5389 "3E = { , 50}, % \geqslant
5390 % "3F = { , }, % \gtrless
5391 "40 = { , 50}, % \sqsubset
5392 "41 = { 50, }, % \sqsupset
5393 "42 = { ,150}, % \vartriangleright, \rhd
5394 "43 = {150, }, % \vartriangleleft, \lhd
5395 "44 = { ,100}, % \trianglerighteq, \unrhd
5396 "45 = {100, }, % \trianglelefteq, \unlhd
5397 "46 = {100,100}, % \bigstar
5398 % "47 = { , }, % \between
5399 "48 = { 50, 50}, % \blacktriangledown
5400 "49 = { ,100}, % \blacktriangleright
5401 "4A = {100, }, % \blacktriangleleft
5402 "4B = { ,150}, % \dashrightarrow (the arrow)
5403 "4C = {150, }, % \dashleftarrow
5404 "4D = { 50, 50}, % \vartriangle
5405 "4E = { 50, 50}, % \blacktriangle
5406 "4F = { 50, 50}, % \triangledown
5407 "50 = { 50, 50}, % \eqcirc
5408 % "51 = { , }, % \lesseqgtr
5409 % "52 = { , }, % \gtreqless
5410 % "53 = { , }, % \lesseqgtr
5411 % "54 = { , }, % \gtreqless
5412 % "55 = { , }, % \yen
5413 "56 = { ,150}, % \Rrightarrow
5414 "57 = {150, }, % \Lleftarrow
5415 "58 = {100,300}, % \checkmark
5416 % "59 = { , }, % \veebar

```

```

5417 % "5A = { , }, % \barwedge
5418 % "5B = { , }, % \doublebarwedge
5419 % "5C = { 50, 50}, % \angle
5420 % "5D = { 50, 50}, % \measuredangle
5421 % "5E = { 50, 50}, % \sphericalangle
5422 % "5F = { , 50}, % \varpropto
5423 % "60 = {100,100}, % \smallsmile
5424 % "61 = {100,100}, % \smallfrown
5425 % "62 = { 50, }, % \Subset
5426 % "63 = { , 50}, % \Supset
5427 % "64 = { , }, % \Cup
5428 % "65 = { , }, % \Cap
5429 % "66 = {150,150}, % \curlywedge
5430 % "67 = {150,150}, % \curlyvee
5431 % "68 = { 50,150}, % \leftthreetimes
5432 % "69 = {100, 50}, % \rightthreetimes
5433 % "6A = { , }, % \subseteqq
5434 % "6B = { , }, % \supseteqq
5435 % "6C = { 50, 50}, % \bumpeq
5436 % "6D = { 50, 50}, % \Bumpeq
5437 % "6E = {100, }, % \lll
5438 % "6F = { ,100}, % \ggg
5439 % "70 = { 50,100}, % \ulcorner
5440 % "71 = {100, 50}, % \urcorner
5441 % "72 = { , }, % \circledR
5442 % "73 = { , }, % \circledS
5443 % "74 = { , }, % \pitchfork
5444 % "75 = {150,200}, % \dotplus
5445 % "76 = { 50,100}, % \backsim
5446 % "77 = { , }, % \backsimeq
5447 % "78 = { 50,100}, % \llcorner
5448 % "79 = {100, 50}, % \lrcorner
5449 % "7A = { , }, % \maltese
5450 % "7B = { , }, % \complement
5451 % "7C = {100,100}, % \intercal
5452 % "7D = { 50, 50}, % \circledcirc
5453 % "7E = { 50, 50}, % \circledast
5454 % "7F = { 50, 50}, % \circleddash
5455 }
5456
5457 </msa>

```

Symbol font 'b'.

```

\DeclareSymbolFont{AMSb}{U}{msb}{m}{n}
\DeclareSymbolFontAlphabet{\mathbb}{AMSb}

```

```

5458 <*msb>
5459 \SetProtrusion
5460 [ name = AMS-b ]
5461 { encoding = U,
5462   family = msb }
5463 {
5464   A = { 50, 50}, % \mathbb
5465   C = { 50, 50},
5466   G = { , 50},
5467   L = { , 50},
5468   P = { , 50},
5469   R = { , 50},
5470   T = { , 50},
5471   V = { 50, 50},
5472   X = { 50, 50},

```

```

5473      Y = { 50, 50},
5474      "00 = { 50, 50}, % \lvertneqq
5475      "01 = { 50, 50}, % \gvertneqq
5476      "02 = { 50, 50}, % \nleq
5477      "03 = { 50, 50}, % \ngeq
5478      "04 = {100, 50}, % \nless
5479      "05 = { 50,150}, % \ngtr
5480      "06 = {100, 50}, % \nprec
5481      "07 = { 50,150}, % \nsucc
5482      "08 = { 50, 50}, % \lneqq
5483      "09 = { 50, 50}, % \gneqq
5484      "0A = {100,100}, % \nleqslant
5485      "0B = {100,100}, % \ngeqslant
5486      "0C = {100, 50}, % \lneq
5487      "0D = { 50,100}, % \gneq
5488      "0E = {100, 50}, % \npreceq
5489      "0F = { 50,100}, % \succeq
5490      "10 = { 50,  }, % \precnsim
5491      "11 = { 50, 50}, % \succnsim
5492      "12 = { 50, 50}, % \lnsim
5493      "13 = { 50, 50}, % \gnsim
5494      "14 = { 50, 50}, % \lneqq
5495      "15 = { 50, 50}, % \ngeqq
5496      "16 = { 50, 50}, % \precneqq
5497      "17 = { 50, 50}, % \succneqq
5498      "18 = { 50, 50}, % \precnapprox
5499      "19 = { 50, 50}, % \succnapprox
5500      "1A = { 50, 50}, % \lnapprox
5501      "1B = { 50, 50}, % \gnapprox
5502      "1C = {150,200}, % \nsim
5503      "1D = { 50, 50}, % \ncong
5504      "1E = {100,150}, % \diagup
5505      "1F = {100,150}, % \diagdown
5506      "20 = {100, 50}, % \varsubsetneq
5507      "21 = { 50,100}, % \varsupsetneq
5508      "22 = {100, 50}, % \subsetneqq
5509      "23 = { 50,100}, % \nsupseteqq
5510      "24 = {100, 50}, % \subsetneqq
5511      "25 = { 50,100}, % \supsetneqq
5512      "26 = {100, 50}, % \varsubsetneqq
5513      "27 = { 50,100}, % \varsupsetneqq
5514      "28 = {100, 50}, % \subsetneq
5515      "29 = { 50,100}, % \supsetneq
5516      "2A = {100, 50}, % \nsubseteq
5517      "2B = { 50,100}, % \nsupseteq
5518      "2C = { 50,100}, % \nparallel
5519      "2D = {100,150}, % \nmid
5520      "2E = {150,150}, % \nshortmid
5521      "2F = {100,100}, % \nshortparallel
5522      "30 = {  ,150}, % \nvdash
5523      "31 = {  ,150}, % \nVdash
5524      "32 = {  ,100}, % \nvDash
5525      "33 = {  ,100}, % \nVDash
5526      "34 = {  ,100}, % \ntrianglerighteq
5527      "35 = {100,  }, % \ntrianglelefteq
5528      "36 = {100,  }, % \ntriangleleft
5529      "37 = {  ,100}, % \ntriangleright
5530      "38 = {100,200}, % \leftarrow
5531      "39 = {100,200}, % \rightarrow
5532      "3A = {100,100}, % \Leftarrow
5533      "3B = { 50,100}, % \Rightarrow

```

```

5534 "3C = {100,100}, % \nLefttrightharpoon
5535 "3D = {100,200}, % \nlefttrightharpoon
5536 "3E = { 50, 50}, % \divideontimes
5537 "3F = { 50, 50}, % \varnothing
5538 % "40 = { , }, % \nexists
5539 "60 = {200, }, % \Finv
5540 "61 = { , 50}, % \Game
5541 % "66 = { , }, % \mho
5542 % "67 = { , }, % \eth
5543 "68 = {100,100}, % \eqsim
5544 "69 = { 50, }, % \beth
5545 "6A = { 50, }, % \gimel
5546 "6B = {150, }, % \daleth
5547 "6C = {200, }, % \lessdot
5548 "6D = { ,200}, % \gtrdot
5549 "6E = {100,200}, % \ltimes
5550 "6F = {150,100}, % \rtimes
5551 "70 = { 50,100}, % \shortmid
5552 "71 = { 50, 50}, % \shortparallel
5553 "72 = {200,300}, % \smallsetminus
5554 "73 = {100,200}, % \thicksim
5555 "74 = { 50,100}, % \thickapprox
5556 "75 = { 50, 50}, % \approxeq
5557 "76 = { 50,100}, % \succapprox
5558 "77 = { 50, 50}, % \precapprox
5559 "78 = {100,100}, % \curvearrowleft
5560 "79 = { 50,150}, % \curvearrowright
5561 "7A = { 50,200}, % \digamma
5562 "7B = {100, 50}, % \varkappa
5563 % "7C = { , }, % \Bbbk
5564 % "7D = { , }, % \hslash
5565 % "7E = { , }, % \hbar
5566 "7F = {200, }, % \backepsilon
5567 }
5568
5569 </msb>

```

Euler Fraktur font (eufrak).

```

\DeclareMathAlphabet{\mathfrak}{U}{euf}{m}{n}
\SetMathAlphabet{\mathfrak}{bold}{U}{euf}{b}{n}

```

```

5570 <*euf>
5571 \SetProtrusion
5572 [ name = mathfrak ]
5573 { encoding = U,
5574   family = euf }
5575 {
5576   A = { , 50},
5577   B = { , 50},
5578   C = { 50, 50},
5579   D = { , 80},
5580   E = { 50, },
5581   G = { , 50},
5582   L = { , 80},
5583   O = { , 50},
5584   T = { , 80},
5585   X = { 80, 50},
5586   Z = { 80, 50},
5587   b = { , 50},
5588   c = { , 50},
5589   k = { , 50},

```



```

5590     p = { , 50},
5591     q = { 50, },
5592     v = { , 50},
5593     w = { , 50},
5594     x = { , 50},
5595     1 = {100,100},
5596     2 = { 80, 80},
5597     3 = { 80, 50},
5598     4 = { 80, 50},
5599     7 = { 50, 50},
5600 }
5601
5602 </euf>

```

Euler script font (euca1).

```

\DeclareMathAlphabet\EuScript{U}{eus}{m}{n}
\SetMathAlphabet\EuScript{bold}{U}{eus}{b}{n}

```

```

5603 <*eus>
5604 \SetProtrusion
5605 [ name = euscript ]
5606 { encoding = U,
5607   family = eus }
5608 {
5609   A = {100,100},
5610   B = { 50,100},
5611   C = { 50, 50},
5612   D = { 50,100},
5613   E = { 50,100},
5614   F = { 50, },
5615   G = { 50, },
5616   H = { ,100},
5617   K = { , 50},
5618   L = { ,150},
5619   M = { , 50},
5620   N = { , 50},
5621   O = { 50, 50},
5622   P = { 50, 50},
5623   T = { ,100},
5624   U = { , 50},
5625   V = { 50, 50},
5626   W = { 50, 50},
5627   X = { 50, 50},
5628   Y = { 50, },
5629   Z = { 50,100},
5630 }
5631
5632 </eus>
5633 </cfg-u>

```

### 14.7.8 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym).

```

5634 <*cfg-e>
5635 \SetProtrusion
5636 <zpeu|euroitc> { encoding = U,
5637 <mvs> { encoding = OT1,
5638 <zpeu> family = zpeu }

```

```

5639 <euroitc>      family   = {euroitc,euroitcs} }
5640 <mvs>          family   = mvs }
5641 {
5642 <zpeu>          E = {50, }
5643 <euroitc>       E = {100,50}
5644 <mvs>          164 = {50,50}    % \EUR
5645 <mvs>          068 = {50,-100}, % \EURdig
5646 }
5647
5648 <*zpeu|euroitc>
5649 \SetProtrusion
5650 <zpeu|euroitc> { encoding = U,
5651 <zpeu>          family   = zpeu,
5652 <euroitc>       family   = {euroitc,euroitcs},
5653      shape      = it* }
5654 {
5655 <zpeu>          E = {100,-50}
5656 <euroitc>       E = {100,}
5657 }
5658
5659 </zpeu|euroitc>
5660 <*zpeu>
5661 \SetProtrusion
5662 { encoding = U,
5663      family   = {zpeus,eurosans} }
5664 {
5665      E = {100,50}
5666 }
5667
5668 \SetProtrusion
5669 { encoding = U,
5670      family   = {zpeus,eurosans},
5671      shape      = it* }
5672 {
5673      E = {200, }
5674 }
5675
5676 </zpeu>
5677 </cfg-e>

```

## 14.8 Interword Spacing

Default unit is space.

```

5678 <*beta>
5679 <*m-t>
5680 %%% -----
5681 %%% INTERWORD SPACING SETTINGS
5682
5683 \SetExtraSpacing
5684 [ name = default ]
5685 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
5686 {

```

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. (. . . In fact, I think these settings are wrong. They lead to more overfull boxes than without spacing adjustment. Needs to be fixed.)

‘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

5687                   {,} = { , -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)

5688                   r = { , -300, 300},

- before or after lowercase characters with ascenders

5689                   b = { , -200, 200},

5690                   d = { , -200, 200},

5691                   f = { , -200, 200},

5692                   h = { , -200, 200},

5693                   k = { , -200, 200},

5694                   l = { , -200, 200},

5695                   t = { , -200, 200},

- before of after lowercase characters with x-height plus descender with additional optical space, e. g., 'v', or 'w'

5696                   c = { , -100, 100},

5697                   p = { , -100, 100},

5698                   v = { , -100, 100},

5699                   w = { , -100, 100},

5700                   z = { , -100, 100},

5701                   x = { , -100, 100},

5702                   y = { , -100, 100}, % ?

- before of after lowercase characters with x-height plus descender without additional optical space

5703                   i = { , 50, -50},

5704                   m = { , 50, -50},

5705                   n = { , 50, -50},

5706                   u = { , 50, -50},

- after colon and semicolon

5707                   : = { , 200, -200},

5708                   ; = { , 200, -200},

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

5709                   . = { , 250, -250},

5710                   ! = { , 250, -250},

5711                   ? = { , 250, -250},

The order has to be reversed when enlarging is needed.'

5712                   }

5713

Questions are:

- Is the result really better?
- Is it overdone? (Try with a factor  $< 1000$ .)
- Should the first parameter also be used? (Probably.)

### 14.8.1 Nonfrenchspacing

The following settings simulate `\nonfrenchspacing` (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the  $\TeX$ book:

‘If the space factor  $f$  is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if  $f \geq 2000$ . [...] Then the stretch component is multiplied by  $f / 1000$ , while the shrink component is multiplied by  $1000 / f$ .’

The ‘extra space’ (`\fontdimen 7`) for Computer Modern Roman is a third of `\fontdimen 2`, i. e., 333.

```
5714 \SetExtraSpacing
5715   [ name      = nonfrench-cmr,
5716     load      = default,
5717     context   = nonfrench ]
5718   { encoding = {OT1,T1,LY1,OT4,QX,T5},
5719     family   = cmr }
5720   {
```

`latex.ltx` has:

```
\def\nonfrenchspacing{
\sfcodes\ . 3000
```

```
5721   . = {333,2000,-667},
```

```
\sfcodes\? 3000
```

```
5722   ? = {333,2000,-667},
```

```
\sfcodes\! 3000
```

```
5723   ! = {333,2000,-667},
```

```
\sfcodes\: 2000
```

```
5724   : = {333,1000,-500},
```

```
\sfcodes\; 1500
```

```
5725   ; = { , 500,-333},
```

```
\sfcodes\, 1250
```

```
5726   {,}= { , 250,-200},
```

```
}
```

```
5727 }
5728
```

fontinst, however, which is also used to create the PSNFSS font metrics, sets \fontdimen 7 to 240 by default. Therefore, the fallback settings use this value for the first component.

```
5729 \SetExtraSpacing
5730 [ name      = nonfrench-default,
5731   load      = default,
5732   context   = nonfrench ]
5733 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
5734 {
5735   . = {240,2000,-667},
5736   ? = {240,2000,-667},
5737   ! = {240,2000,-667},
5738   : = {240,1000,-500},
5739   ; = {      , 500,-333},
5740   {,}= {      , 250,-200},
5741 }
5742
```

## 14.9 Additional Kerning

Default unit is 1em.

```
5743 %%% -----
5744 %%% ADDITIONAL KERNING
5745
```

A dummy list to be loaded when no context is active.

```
5746 \SetExtraKerning
5747 [ name = empty ]
5748 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
5749 { }
5750
```

### 14.9.1 French

For the French context.

```
5751 \SetExtraKerning
5752 [ name      = french-default,
5753   context   = french,
5754   unit      = space ]
5755 { encoding = {OT1,T1,LY1} }
5756 {
5757   : = {1000,}, % = \fontdimen2
5758   ; = {500, }, % ~ \thinspace
5759   ! = {500, },
5760   ? = {500, },
5761 }
5762
```

This has the disadvantage that the word following a left guillemet will not be hyphenated. This might be fixed in pdf<sub>T</sub><sub>E</sub>X.

```
5763 \SetExtraKerning
```

```

5764 [ name      = french-guillemets,
5765     context  = french-guillemets,
5766     load     = french-default,
5767     unit     = space ]
5768 { encoding = {OT1,T1,LY1} }
5769 {
5770   \guillemotleft = { ,800}, % = 0.8\fontdimen2
5771   \guillemotright = {800, },
5772 }
5773

```

### 14.9.2 Turkish

And for Turkish.

```

5774 \SetExtraKerning
5775 [ name      = turkish,
5776     context  = turkish,
5777     unit     = space ]
5778 { encoding = {OT1,T1,LY1} }
5779 {
5780   : = {500, }, % ~ \thinspace
5781   ! = {500, },
5782   {=} = {500, },
5783 }
5784

```

### 14.9.3 Letterspacing

The settings with the ‘letterspacing’ context will be loaded whenever the command `\textls` resp. `\lsstyle` are used.

```

5785 %%% The following settings with `context=letterspacing'
5786 %%% will be loaded by \lsstyle and \textls:
5787

```

No additional kerning will be applied in math mode, but we don’t want a confusing error messages (e. g., when writing L<sup>A</sup>T<sub>E</sub>X).

```

5788 \SetExtraKerning
5789 [ name      = letterspacing-all,
5790     context  = letterspacing,
5791     preset   = {1000,1000} ]
5792 { encoding = {U,OML,OMS,OMX} }
5793 { }
5794

```

The full stop should be spaced out less. Numbers are not spaced out, according to soul.

```

5795 \SetExtraKerning
5796 [ name      = letterspacing-text,
5797     context  = letterspacing,
5798     preset   = {1000,1000} ]
5799 { }
5800 {
5801   . = {0, },
5802   0 = {0,0},
5803   1 = {0,0},
5804   2 = {0,0},
5805   3 = {0,0},
5806   4 = {0,0},

```

```

5807     5 = {0,0},
5808     6 = {0,0},
5809     7 = {0,0},
5810     8 = {0,0},
5811     9 = {0,0},
5812   }
5813

```

Also, quotation marks receive only half the kerning.

```

5814 \SetExtraKerning
5815   [ name      = letterspacing-OT1,
5816     context   = letterspacing,
5817     load      = letterspacing-text,
5818     preset    = {1000,1000} ]
5819   { encoding = {OT1,OT4,QX} }
5820   {
5821     \textquoteleft   = {0,0},   \textquoteright   = {0,0},
5822     \textquotedblleft = {0,0},   \textquotedblright = {0,0},
5823   }
5824
5825 \SetExtraKerning
5826   [ name      = letterspacing-T1,
5827     context   = letterspacing,
5828     load      = letterspacing-text,
5829     preset    = {1000,1000} ]
5830   { encoding = {T1,L1,T5} }
5831   {
5832     \textquoteleft   = {0,0},   \textquoteright   = {0,0},
5833     \textquotedblleft = {0,0},   \textquotedblright = {0,0},
5834     \quotesinglbase   = {0,0},   \quotedblbase   = {0,0},
5835     \guilsinglleft    = {0,0},   \guilsinglright  = {0,0},
5836     \guillemotleft    = {0,0},   \guillemotright  = {0,0},
5837   }
5838
5839 /m-t
5840 /beta
5841 /config

```

## 15 Auxiliary File for Micro Fine Tuning

This file can be used to test protrusion and expansion settings.

```

5842 *test
5843 \documentclass{article}
5844
5845 %% Here you can specify the font you want to test, using
5846 %% the commands \fontfamily, \fontseries, and \fontshape.
5847 %% Make sure to end all lines with a comment character!
5848 \newcommand*{\TestFont{%
5849   \fontfamily{ppl}%
5850   \fontseries{b}%
5851   \fontshape{it}% sc, sl
5852 }
5853
5854 \usepackage{ifthen}
5855 \usepackage[T1]{fontenc}
5856 \usepackage[latin1]{inputenc}
5857 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
5858
5859 \pagestyle{empty}

```

```

5860 \setlength{\parindent}{0pt}
5861 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
5862 \newcommand*\testprotrusion[2][\%
5863   \ifthenelse{\equal{#1}{r}}{\#2}%
5864   lorem ipsum dolor sit amet,
5865   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
5866   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
5867   you know the rest%
5868   \ifthenelse{\equal{#1}{l}}{\#2}%
5869   \linebreak
5870   {\fontencoding{\encodingdefault}%
5871   \fontseries{\seriesdefault}%
5872   \fontshape{\shapedefault}%
5873   \selectfont
5874   Here is the beginning of a line, \dotfill and here is its end}\linebreak
5875 }
5876 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
5877 \def\stripprefix#1>{}
5878 \newcount\charcount
5879 \begin{document}
5880
5881 \microtypesetup{expansion=false}
5882
5883 {\centering The font in this document is called by:\\
5884 \texttt{\showTestFont}\par}\bigskip
5885
5886 \TestFont\selectfont
5887 This line intentionally left empty\linebreak
5888 %% A -- Z
5889 \charcount=65
5890 \loop
5891   \testprotrusion{\char\charcount}
5892   \advance\charcount 1
5893   \ifnum\charcount < 91 \repeat
5894 %% a -- z
5895 \charcount=97
5896 \loop
5897   \testprotrusion{\char\charcount}
5898   \advance\charcount 1
5899   \ifnum\charcount < 123 \repeat
5900 %% 0 -- 9
5901 \charcount=48
5902 \loop
5903   \testprotrusion{\char\charcount}
5904   \advance\charcount 1
5905   \ifnum\charcount < 58 \repeat
5906 %%
5907 \testprotrusion[r]{,}
5908 \testprotrusion[r]{.}
5909 \testprotrusion[r]{;}
5910 \testprotrusion[r]{:}
5911 \testprotrusion[r]{?}
5912 \testprotrusion[r]{!}
5913 \testprotrusion[l]{\textexclamdown}
5914 \testprotrusion[l]{\textquestiondown}
5915 \testprotrusion[r]{})}
5916 \testprotrusion[l]{(}
5917 \testprotrusion{/}
5918 \testprotrusion{\char~\}
5919 \testprotrusion{-}
5920 \testprotrusion{\textendash}

```



```

5921 \testprotrusion{\textemdash}
5922 \testprotrusion{\textquoteleft}
5923 \testprotrusion{\textquoteright}
5924 \testprotrusion{\textquotedblleft}
5925 \testprotrusion{\textquotedblright}
5926 \testprotrusion{\quotesinglbase}
5927 \testprotrusion{\quotedblbase}
5928 \testprotrusion{\guilsinglleft}
5929 \testprotrusion{\guilsinglright}
5930 \testprotrusion{\guillemotleft}
5931 \testprotrusion{\guillemotright}
5932
5933 \newpage
5934 The following displays the current font stretched by 5%,
5935 normal, and shrunk by 5%:
5936
5937 \bigskip
5938 \newlength{\MTln}
5939 \newcommand*{\teststring}
5940 {ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
5941 \settowidth{\MTln}{\teststring}
5942 \microtypesetup{expansion=true}
5943
5944 \parbox{1.05\MTln}{\teststring\linebreak\
5945 \teststring}\par\bigskip
5946 \parbox{0.95\MTln}{\teststring}
5947
5948 \end{document}
5949 /test

```

Needless to say that things may always be improved. For suggestions, mail to [w.m.l@gmx.net](mailto:w.m.l@gmx.net).

## A Change History

### Version 1.0 (2004/09/11)

General: Initial version . . . . . 1

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

General: configuration file names in lowercase (suggested by <i>Harald Harders</i> ) . . . . .	57	command: possibility to specify character inheritance . . . . .	83
issue an error instead of a warning, when pdf <sub>T</sub> <sub>E</sub> X version is too old for autoexpand . . . . .	99	\MT@declare@sets: remove spaces around set name . . . . .	71
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i> ) . . . . .	105	\MT@DeclareSet@: remove spaces around first argument . . . . .	71
Protrusion: add factors for some more characters . . . . .	110	\MT@find@file: fix: also check whether the file for the base font family has already been loaded . . . . .	57
settings for Adobe Minion (contributed by <i>Harald Harders</i> ) . . .	111	\MT@get@basefamily: only remove suffix, if it is 'x' or 'j' . . . . .	58
\DeclareCharacterInheritance: new		\MT@get@listname@: don't check for empty attributes list . . . . .	58
		\MT@ifempty: fix: use category code 12	

for the percent character (reported by <i>Tom Kink</i> ) . . . . .	31	\MT@permute: don't use sets for empty encoding . . . . .	85
\MT@is@number: numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i> ) . .	62	\MT@pr@split: fix: allow zero and negative values . . . . .	44
\MT@pdftex@no: fix concerning version check (reported by <i>Harald Harders</i> ) . . . . .	26	\MT@use@set: remove spaces around set name . . . . .	75
		\UseMicrotypeSet: remove spaces around first argument . . . . .	75

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

General: check for packages that might load fonts . . . . .	66	faults have changed . . . . .	72
check whether only one encoding specified . . . . .	84	\MT@get@listname@: alternatively check for alias font name . . . . .	58
Font Sets: declare cmr as an alias of cmr new: allmath and basicmath . . .	103	\MT@get@size: additional magic to catch some errors . . . . .	74
Protrusion: add settings for Adobe Garamond and Computer Modern Roman in TS1 encoding . . . . .	131	hijack \set@fontsize instead of \setfontsize . . . . .	73
add settings for Computer Modern Roman math symbols . . . . .	136	\MT@get@slot: fix: group must also include \MT@get@composite . . . . .	61
\MT@context: fix: set inheritance list globally to \empty . . . . .	60	\MT@loop: fix: new macro, used instead of \loop . . . . .	34
\MT@familyalias: define alias font name as an alternative, not as a replacement . . . . .	40	\MT@maybe@do: also check for alias font name . . . . .	40
\MT@get@basefamily: also remove 'w' (swash capitals) . . . . .	58	\MT@permute@@@@: more sanity checks for \SetProtrusion and \SetExpansion . . . . .	86
\MT@get@highlevel: check whether de-		\MT@setupfont: also search for alias font file . . . . .	39
		fix: call \@enc@update if necessary	40

### Version 1.3 (2004/10/27)

Font Sets: declare aer, zer and hfor as an alias of cmr . . . . .	104	\MT@get@codes@name: fix: specifying load option does no longer require to give a name, too . . . . .	80
\MT@catcodes: check some category codes (compatibility with german)	27	\MT@load@list: check whether list exists	57

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

General: don't use scratch registers in global definitions . . . . .	61	cmr variants (OT1, T1, lmr) . . .	115
no need to check for packages that might load fonts anymore . . . .	66	\microtypesetup: fix: set the correct levels, and remember them; warning when enabling an option disabled in package options . . . . .	94
use \pickup@font instead of \define@newfont as the hook for \MT@setupfont . . . . .	66	\MT@pdfcp@error: check for pdfcp@	37
use one instead of five counters . .	35	\SetExpansion: fix: specifying extra options does no longer require to give a name, too . . . . .	78
Protrusion: tweak quote characters for			

**Version 1.4a (2004/11/17)**

General: new option: final . . . . .	91	more catcodes when reading files (reported by <i>Michael Hoppe</i> ) . . .	58
\MT@begin@catcodes: fix: reset some			

**Version 1.4b (2004/11/26)**

General: fix: set catcodes before reading global configuration file (reported by <i>Christoph Bier</i> ) . . . . .	93	alias family name if encoding failed . . . . .	42
new message if \pdfoutput is changed . . . . .	97	\MT@get@basefamily: fix: failed for font names of the form abczz (reported by <i>Georg Verweyen</i> ) . . . . .	58
optimization: use less \csnames and \expandafters . . . . .	29	\MT@get@slot: don't define \MT@char globally (save stack problem) . . .	61
Protrusion: harmonize dashes in upshape and italic (cmr, pad, ppl) . . . . .	110	\MT@ifdimen: don't set \MT@count glob- ally (save stack problem) . . . . .	31
slanted like italics . . . . .	118	\MT@use@set: don't use undeclared font sets . . . . .	75
\MT@checklist@family: fix: don't try			

**Version 1.5 (2004/12/15)**

General: defaults: step: 4 (suggested by <i>Hàn Thế Thành</i> ) . . . . .	91	\DeclareMicrotypeAlias: remove spaces around arguments . . . . .	76
defaults: calculate step as min(stretch,shrink)/5 . . . . .	98	\MT@begin@catcodes: reset catcode of '=' (compatibility with Turkish babel) . . .	58
defaults: turn off expansion for DVI output . . . . .	97	\MT@catcodes: reset catcode of '^' (com- patibility with chemsym) . . . . .	27
disable automatic expansion for DVI output . . . . .	98	\MT@get@highlevel: don't test defaults if called after begin document . . . . .	72
new option: selected, by default false (suggested by <i>Hàn Thế Thành</i> ) . . . . .	90	\MT@scale@factor: warning for factors outside limits . . . . .	46
Documentation: add note about DVIoutput option . . . . .	7	\MT@scale@to@em: don't use \lcode and \rcode for the calculation . . . . .	44
add short history (section 12) . . . . .	21	\MT@set@ex@codes: allow non-selected font expansion . . . . .	49
Inheritance: remove \ss from T1 list, add \DJ . . . . .	106	\MT@set@pr@codes: adjust protrusion fac- tors before setting the inheriting characters . . . . .	43
Protrusion: settings for Bitstream Char- ter . . . . .	111		

**Version 1.6 (2005/01/24)**

General: defaults: turn off expansion for old pdf $\TeX$ versions . . . . .	92	use e- $\TeX$ 's \ifcsname and \ifdefined if defined . . . . .	30
disable automatic expansion for old pdf $\TeX$ versions . . . . .	99	Protrusion: add italic uppercase Greek letters . . . . .	118
load a font, if none is active . . . . .	39	improve settings for numbers (pointed out by <i>Peter Muthesius</i> ) . . .	112
new option: factor, by default 1000 . . . . .	91	tune CMR math letters (OML encod- ing) . . . . .	136
restructure dtx file . . . . .	103	\MT@def@num@opt: test whether numeric	
test whether \pickup@font has changed . . . . .	68		

options receive a number . . . . .	91	\MT@set@pr@codes: introduce factor op-	
\MT@get@charwd: use e-TeX's		tion . . . . .	43
\fontcharwd, if available . . . . .	45	\MT@use@set: retain current set if new	
\MT@get@inh@list: correct message if		set is undeclared . . . . .	75
selected is false . . . . .	60	\MT@vinfo: new macro: used instead of	
\MT@set@ex@codes: introduce factor op-		\ifMT@verbose . . . . .	25
tion . . . . .	49		

### Version 1.6a (2005/02/02)

Documentation: add table of fonts with		<i>Bernard Gaulle</i> ) . . . . .	61
tailored protrusion settings . . .	15	\MT@pdf@tex@no: new macro . . . . .	26
\MT@get@slot: completely redone, hope-		\MT@reset@ef@codes: only reset \efcodes	
fully more robust (compatible with		for older pdfTeX versions . . . . .	49
frenchpro; problem reported by			

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

General: \SetExpansion: bug fix: remove		test for \chardefed commands . . .	61
space after autoexpand . . . . .	83	test whether \langle encoding \rangle is de-	
\SetExpansion: don't allow auto-		defined . . . . .	61
matic expansion for old pdfTeX		\MT@if@list@exists: don't define	
versions . . . . .	83	\MT@#1@c@name \globally, here	
allow specification of size ranges		and elsewhere . . . . .	60
(suggested by <i>Andreas Böhmann</i> )	72	\MT@if@dimen: comparison with 1 to allow	
modify \showhyphens . . . . .	99	size smaller than 1 (suggested by	
new value for verbose option:		<i>Andreas Böhmann</i> ) . . . . .	31
errors . . . . .	91	\MT@increment: use e-TeX's \numexpr if	
shorter command names . . . . .	35	available . . . . .	35
warning when running in draft mode	96	\MT@is@composite: new macro: construct	
Documentation: add hint about compati-		command for composite character;	
bility . . . . .	18	no uncontrolled expansion . . .	65
remove table of match order . . . .	11	\MT@scale: new macro: use e-TeX's	
Protrusion: fix: remove \ from OT1, add		\numexpr if available . . . . .	35
\textbackslash to T1 encoding	113	\MT@set@ex@codes: two versions of this	
\DeclareMicrotypeAlias: may also be		macro . . . . .	49
used inside configuration files .	76	\MT@split@name: don't define	
\LoadMicrotypeFile: new command		\MT@encoding &c. \globally . . .	40
(suggested by <i>Andreas Böhmann</i> )	76	\MT@test@ast: make it simpler . . . . .	72
\Microtype@Hook: new command for font		\MT@try@order: always check for size, too	
package authors . . . . .	93	(suggested by <i>Andreas Böhmann</i> )	59
\microtypesetup: fix: warning also when		fix: also check for	
setting to (no)compatibility . .	94	//\series/\shape// (reported	
\MT@begin@catcodes: also use inside con-		by <i>Andreas Böhmann</i> ) . . . . .	59
figuration commands . . . . .	58	\MT@warn@code@too@large: new macro:	
reset catcode of ':' (compatibility		type out maximum protrusion fac-	
with french* packages) . . . . .	58	tor . . . . .	46
\MT@get@listname@: use \etfor ( <i>Andreas</i>		\MT@warn@err: new macro: for	
<i>Böhmann's</i> idea) . . . . .	58	verbose=errors . . . . .	25
\MT@get@slot: remove backslash hack	61		

**Version 1.8 (2005/06/23)**

General: <code>\SetProtrusion</code> : new option: unit . . . . .	81	<code>\MT@get@listname@</code> : made recursive . .	58
if font substitution has occurred, set up the substitute font, not the se- lected one . . . . .	66	<code>\MT@get@slot</code> : fix: expand active charac- ters . . . . .	61
new option: config to load a differ- ent main configuration file . . . . .	92	test whether <code>\langle encoding \rangle \langle ... \rangle</code> is de- fined made more robust . . . . .	61
new option: unit, by default character . . . . .	92	<code>\MT@get@unit</code> : new macro: get unit for codes . . . . .	47
Documentation: add example for factor option . . . . .	12	<code>\MT@in@rlist</code> : made recursive . . . . .	34
add example of how to get rid of a widow (suggested by <i>Adam Ku-</i> <i>charczyk</i> ) . . . . .	13	<code>\MT@is@active</code> : new macro: translate inputenc-defined characters . . .	63
add hint about error messages . . .	19	<code>\MT@is@letter</code> : warning for non-ASCII characters . . . . .	62
Font Sets: add U encoding to <code>allmath</code>	103	<code>\MT@led@kern</code> : character protrusion with <code>ledmac</code> . . . . .	37
declare <code>pxr</code> and <code>txr</code> as aliases of <code>ppl</code> <code>resp. ptm</code> . . . . .	104	<code>\MT@make@string</code> : use <code>\@onelevel@sanitize</code> . . . . .	35
Inheritance: remove <code>\DJ</code> from T1 list (it's the same as <code>\DH</code> ) . . . . .	106	<code>\MT@map@clist@n</code> : new macro: used in- stead of <code>\@for</code> . . . . .	33
Protrusion: add LY1 characters for Times	117	<code>\MT@map@tlist@n</code> : new macro: used in- stead of <code>\@tfor</code> . . . . .	33
settings for AMS math fonts . . . .	140	<code>\MT@old@cmd</code> : renamed commands from <code>\..MicroType..</code> to <code>\..Microtype..</code> . . . . .	25
verified settings for slanted Com- puter Modern Roman . . . . .	124	<code>\MT@orig@add@accent</code> : fix: disable micro-typographic setup inside <code>\add@accent</code> (reported by <i>Stephan</i> <i>Hennig</i> ) . . . . .	67
<code>\DeclareMicrotypeAlias</code> : warning when overriding an alias font . . . . .	76	<code>\MT@pdf@tex@no</code> : case 5: pdf $\TeX$ 1.30 . .	26
<code>\DeclareMicrotypeSetDefault</code> : new command: set default font set . .	75	<code>\MT@permute@</code> : add ranges to the beginning of the lists . . . . .	86
<code>\MT@begin@catcodes</code> : reset catcodes of the remaining ASCII characters . . .	58	<code>\MT@pr@split</code> : get character width once only . . . . .	44
<code>\MT@check@rlist</code> : made recursive . . .	87	<code>\MT@scale</code> : fix: remove spaces in non- e- $\TeX$ variant (reported by <i>Mark</i> <i>Rossi</i> ) . . . . .	35
<code>\MT@curr@list@name</code> : new macro: cur- rent list type and name . . . . .	65	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>hyperref</code> is loaded . . . . .	38
<code>\MT@declare@sets</code> : warning when re- defining a set . . . . .	71	restore <code>csquotes</code> 's active characters restore percent character if Spanish <code>babel</code> is loaded . . . . .	38
<code>\MT@define@set@key@</code> : use comma lists instead of token lists . . . . .	71	<code>\MT@use@set</code> : fix: remove braces in first line . . . . .	75
<code>\MT@find@file</code> : no longer wrap names in commands . . . . .	57	<code>\MT@xadd</code> : simplified . . . . .	32
<code>\MT@get@charwd</code> : warning for missing (resp. zero-width) characters . .	45		
<code>\MT@get@dimen@six</code> : new macro: test whether <code>\fontdimen 6</code> is defined . .	43		

**Version 1.9 (2005/10/28)**

General: <code>\DeclareMicrotypeSet</code> : new key: font . . . . .	74	<code>\SetProtrusion</code> : value 'relative' re- named to 'character' for option unit . . . . .	81
<code>\SetProtrusion</code> and <code>\SetExpansion</code> : new key: font . . . . .	80	allow context-specific font setup . .	66

disable expansion if both step and shrink are zero . . . . .	99	1.30) . . . . .	77
disable microtype setup inside hyperref's \pdfstringdef (reported by <i>Hàn Thế Thành</i> ) . . . .	38	\microtypecontext: new command: change setup context in the document . . . . .	69
option unit: rename value relative to character . . . . .	92	\MT@checklist@family: fix: add two missing \expandafters . . . . .	42
warning if user requested zero step . . . . .	98	\MT@define@option: fix: use true as the default value . . . . .	88
Documentation: add hint about verbatim environment . . . . .	18	\MT@detokenize@c: fix the non-e-TeX version . . . . .	31
add remark about Type 1 fonts required for automatic font expansion . . . . .	6	\MT@exp@two@n: new macros: less \expandafters . . . . .	29
Font Sets: add OT4 encoding to text sets . . . . .	103	\MT@get@opt: new key 'preset' to set all characters to the specified value before loading the lists . . . . .	47
add T5 encoding to text sets . . . . .	103	\MT@is@active: redone: use \set@display@protect . . . . .	63
declare qpl and qtm (qfonts) as aliases of ppl resp. ptm . . . . .	104	\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning . . . . .	62
Inheritance: add list for OT4 . . . . .	107	\MT@maybe@do: redone . . . . .	40
add list for T5 (requested by <i>Hàn Thế Thành</i> ) . . . . .	108	\MT@pdfTeX@no: compatibility with TeXLive hack (reported by <i>Herbert Voß</i> ) . . . . .	26
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR . . . . .	114	\MT@rem@from@clist: new macro: remove an item from a comma list . . . . .	33
settings for OT4 encoding (Computer Modern Roman, Palatino, Times) . . . . .	110	\MT@scale@factor: generalized . . . . .	46
settings for T5 encoding (Computer Modern Roman) . . . . .	110	\MT@toks: use instead of \toks@ . . . . .	28
\DisableLigatures: new command: disable ligatures (requires pdfTeX			

### Version 1.9a (2005/12/05)

General: new option: defersetup, by default true . . . . .	90	\MT@get@codes@name: '(file name)/(line number)' as default list name . . . . .	79
remove superfluous test whether \pickup@font has changed . . . . .	68	\MT@get@highlevel: no longer check whether defaults have changed . . . . .	72
Documentation: add explanation for error message in DVI mode . . . . .	19	\MT@ifdefined@c@T: new macros: TRUE case only . . . . .	30
add explanation for error message with bitmap fonts . . . . .	19	\MT@ifint: use \pdfmatch if available . . . . .	31
Font Sets: declare mdbch (mathdesign) as an alias of Charter . . . . .	104	\MT@ifstreq: use \pdfstrcmp if available . . . . .	32
Protrusion: fix: remove _ from OT1 encoding . . . . .	114	\MT@in@clist: fix . . . . .	33
settings for T5 encoded Charter . . . . .	110	\MT@info@missing@char: info instead of warning (after <i>Michael Hoppe</i> reported that the 'fl' ligature is missing in Palatino SC) . . . . .	45
\microtypesetup: inside the preamble, accepts all package options . . . . .	94	\MT@is@feature: new macro: check for pdfTeX feature . . . . .	36
\MT@check@font@cx: optimize context-sensitive setup . . . . .	68	\MT@map@clist@n: following L <sup>A</sup> T <sub>E</sub> X3 . . . . .	33
\MT@define@set@key@: don't expand variables immediately (requested by <i>Georg Verweyen</i> ) . . . . .	71	\MT@permute@@@@: don't define permutations for unused encodings . . . . .	85
		\MT@rem@from@clist: fix . . . . .	33
		\MT@setup@: defer setup until the end of the preamble . . . . .	36

**Version 1.9b (2006/01/20)**

General: compatibility with the listings package (reported by <i>Holger Uhr</i> )	39	add samples of micro-typographic features	4
compatibility with the extendedchar option of the listings package	39	\MT@features: use throughout the package to adjust to beta-ness	36
register with the soul package	39	\MT@ifdimen: use \pdfmatch if available	31
Documentation: activate expansion in the distributed PDF	1	\MT@warn@code@too@large: fix calculation with present factor	46

**Version 1.9c (2006/02/02)**

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verweyen</i> )	16	\MT@define@code@key@font: fix: context was ignored	80
Protrusion: settings for URW Garamond	111	\MT@define@code@key@size: fix: embrace \MT@tempsize in \csname (bug introduced in 1.9b)	80

**Version 1.9d (2006/05/05)**

Font Sets: md* instead of m series in basic sets	103	\MT@get@ex@opt: fix: evaluate preset	50
add QX encoding to text sets	103	\MT@get@font@dimen: warning for zero fontdimen	45
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i> )	107	\MT@get@opt: optimize: don't reset when preset option is set	47
Protrusion: settings for QX encoding (contributed by <i>Maciej Eder</i> )	116	set list name before presetting	47
settings for Euro symbols (Adobe, ITC, Marvosym)	145	\MT@is@active: support for Unicode (inputenc/utf8)	64
tweak AMS settings	140	\MT@setupfont@hook: restore \% and \# when tex4ht is loaded (reported by <i>Peter Dybala</i> )	38
\DeclareCharacterInheritance: fix: empty context	83	\SetProtrusion: (et al.) optimize: unify keys for mandatory argument	77
\lststyle: fix: font was always added to list	55	(et al.) split keys of optional and mandatory argument	77
\MT@detokenize@n: new macro: use \detokenize if available	31		

**Version 2.0 ()**

General: new option: babel, by default false (language-dependant setup suggested by <i>Ulrich Dirr</i> )	90	\MT@pdfTeX@no: case 6: pdfTeX 1.40	26
new option: letterspacing, by default 100	91	\SetExtraKerning: new command: additional kerning	79
\DeclareMicrotypeBabelHook: new command: interaction with babel	77	\SetExtraSpacing: new command: adjustment of interword spacing	79
		\textls: new command: letterspacing	56



## B Index

Links refer to the page where the corresponding entry is described (**bold italic**) resp. occurs. Plain numbers refer to the code line where the corresponding entry is defined (underlined) resp. used.

### Options:

DVIoutput	<a href="#">7</a>
activate	<a href="#">5</a>
auto	<a href="#">6</a>
config	<a href="#">8</a>
draft	<a href="#">8</a>
expansion	<a href="#">5</a>
factor	<a href="#">6</a>
final	<a href="#">8</a>
protrusion	<a href="#">5</a>
selected	<a href="#">7</a>
shrink	<a href="#">7</a>
step	<a href="#">7</a>
stretch	<a href="#">7</a>
unit	<a href="#">6</a>
verbose	<a href="#">8</a>

### User Commands:

\DeclareCharacterInheritance	<a href="#">14</a>
\DeclareMicrotypeAlias	<a href="#">15</a>
\DeclareMicrotypeSet	<a href="#">8</a>
\DeclareMicrotypeSet*	<a href="#">8</a>
\DeclareMicrotypeSetDefault	<a href="#">10</a>
\DisableLigatures	<a href="#">17</a>
\LoadMicrotypeFile	<a href="#">16</a>
\microtypecontext	<a href="#">16</a>
\microtypesetup	<a href="#">8</a>
\SetExpansion	<a href="#">13</a>
\SetProtrusion	<a href="#">11</a>
\UseMicrotypeSet	<a href="#">10</a>

\@@enc@update	<a href="#">628</a>
---------------	---------------------

### A

a0poster (package)	<a href="#">74</a>
activate (option)	<a href="#">5</a> , <a href="#">89</a>
\add@accent	<a href="#">1769</a> , <a href="#">1770</a>
ae (package)	<a href="#">16</a> , <a href="#">104</a>
amssymb (package)	<a href="#">140</a>
article (package)	<a href="#">17</a>
auto (option)	<a href="#">6</a> , <a href="#">90</a>

### B

babel (option)	<a href="#">90</a> , <a href="#">159</a>
babel (package)	<a href="#">2</a> , <a href="#">3</a> , <a href="#">22</a> , <a href="#">38</a> , <a href="#">58</a> , <a href="#">77</a> , <a href="#">101</a> , <a href="#">102</a> , <a href="#">104</a> , <a href="#">155</a> , <a href="#">157</a> , <a href="#">159</a>

### C

\cf@encoding	<a href="#">628</a>
chemsym (package)	<a href="#">155</a>
CJK (package)	<a href="#">18</a> , <a href="#">68</a>
cm-super (package)	<a href="#">6</a>
color (package)	<a href="#">7</a> , <a href="#">97</a>
config (option)	<a href="#">8</a> , <a href="#">15</a> , <a href="#">21</a> , <a href="#">92</a> , <a href="#">157</a>
contour (package)	<a href="#">97</a>
crop (package)	<a href="#">97</a>
csquotes (package)	<a href="#">20</a> , <a href="#">22</a> , <a href="#">38</a> , <a href="#">114</a> , <a href="#">157</a>
\curr@fontshape	<a href="#">1345</a> , <a href="#">1350</a> , <a href="#">1740</a>

### D

\DeclareCharacterInheritance	<a href="#">9</a> , <a href="#">14</a> , <a href="#">2447</a> , <a href="#">3062</a>
\DeclareMicrotypeAlias	<a href="#">5</a> , <a href="#">15</a> , <a href="#">33</a> , <a href="#">2140</a> , <a href="#">3063</a>
\DeclareMicrotypeBabelHook	<a href="#">16</a> , <a href="#">25</a> , <a href="#">2188</a>
\DeclareMicrotypeSet	<a href="#">2</a> , <a href="#">8</a> , <a href="#">20</a> , <a href="#">21</a> , <a href="#">31</a> , <a href="#">1870</a> , <a href="#">3054</a>
\DeclareMicrotypeSet*	<a href="#">8</a> , <a href="#">1870</a>
\DeclareMicrotypeSetDefault	<a href="#">4</a> , <a href="#">10</a> , <a href="#">23</a> , <a href="#">2112</a>
defersetup (option)	<a href="#">38</a> , <a href="#">90</a> , <a href="#">94</a> , <a href="#">158</a>
\define@newfont	<a href="#">1742</a>
\DisableLigatures	<a href="#">8</a> , <a href="#">17</a> , <a href="#">24</a> , <a href="#">2173</a>
\do@subst@correction	<a href="#">1739</a>
docstrip (package)	<a href="#">24</a>
draft (option)	<a href="#">8</a> , <a href="#">90</a> , <a href="#">91</a>
dsfont (package)	<a href="#">43</a>
DVIoutput (option)	<a href="#">7</a> , <a href="#">21</a> , <a href="#">90</a> , <a href="#">155</a>

### E

eco (package)	<a href="#">16</a> , <a href="#">104</a>
\efcode	<a href="#">1046</a> , <a href="#">1080</a> , <a href="#">1081</a> , <a href="#">1125</a> , <a href="#">1128</a>
eucal (package)	<a href="#">145</a>
eufrak (package)	<a href="#">144</a>
euroitc (package)	<a href="#">145</a>
europs (package)	<a href="#">145</a>
eurosans (package)	<a href="#">145</a>
expansion (option)	<a href="#">5</a> , <a href="#">6</a> , <a href="#">10</a> , <a href="#">88</a> , <a href="#">89</a> , <a href="#">97</a>

### F

\f@encoding	<a href="#">628</a>
\f@size	<a href="#">1345</a> , <a href="#">1350</a> , <a href="#">1740</a>
factor (option)	<a href="#">6</a> , <a href="#">12</a> , <a href="#">22</a> , <a href="#">91</a> , <a href="#">92</a> , <a href="#">155</a>
fancyvrb (package)	<a href="#">19</a>



- final (option) . . . . . 8, 22, 90, 91, 155  
 fix-cm (package) . . . . . 13  
 \font . . . . . 619  
 \font@name . . . . .  
     856, 1742, 1754, 1756, 1777, 1797  
 \fontcharwd . . . . . 835  
 \fontdimen . 777, 784, 855, 860, 965, 968  
 fontinst (package) . . . . . 149  
 fontinstallationguide (package) .. 129  
 french (package) . . . . . 156  
 frenchpro (package) . . . . . 22, 64, 156
- G**
- german (package) . . . . . 22, 27, 154  
 graphics (package) . . . . . 7, 97
- H**
- hfoldsty (package) . . . . . 16, 104  
 hyperref (package) 7, 38, 66, 97, 157, 158
- I**
- IEEEtran (package) . . . . . 66  
 \iffontchar . . . . . 865  
 \ifMT@auto . . . . . 123, 3128, 3175  
 \ifMT@babel . . . . . 123, 3257, 3280  
 \ifMT@do . . . . . 647, 730, 1393  
 \ifMT@document . . . . . 134, 2561  
 \ifMT@draft . . . . . 123, 3042  
 \ifMT@expansion . . . . . 123, 3092, 3172  
 \ifMT@inlist@ 380, 431, 504, 675, 697,  
     714, 723, 743, 1347, 1426, 1437,  
     1525, 1761, 1785, 1848, 1855, 2158  
 \ifMT@kerning 123, 1340, 3217, 3244, 3305  
 \ifMT@noligatures . . . . . 123, 1382  
 \ifMT@nonselected . . . 1027, 1538, 1547  
 \ifMT@noreset . . . . . 1599, 1606  
 \ifMT@opt@auto . . . . . 2685, 3131, 3143  
 \ifMT@opt@DVI . . . . . 2685, 3074  
 \ifMT@opt@expansion . . . . . 2685, 3069  
 \ifMT@protrusion . . . . . 123, 531, 3075  
 \ifMT@selected . . . . . 123, 3158, 3178  
 \ifMT@spacing .. 123, 3202, 3236, 3256  
 ifpdf (package) . . . . . 7, 97  
 inputenc (package) . . . . .  
     .. 18, 21, 39, 61, 63–65, 157, 159
- J**
- jurabib (package) . . . . . 66
- K**
- kerning (option) . . . . . 89  
 keyval (package) . . . . . 28, 44, 71, 105  
 \knaccode . 1289, 1290, 1302, 1306, 1313  
 \knbccode . 1282, 1283, 1301, 1305, 1312
- \knbscode . 1175, 1176, 1201, 1206, 1214
- L**
- ledmac (package) ... 20, 22, 37, 66, 157  
 ledpar (package) . . . . . 20  
 \leftmarginkern . . . . . 540  
 letterspacing (option) . . . . . 91, 159  
 lineno (package) . . . . . 37  
 listings (package) .. 19, 21, 39, 58, 159  
 lmodern (package) . . . . . 6, 115  
 \LoadMicrotypeFile 10, 16, 34, 2154, 3064  
 \lpcode . . . . . 770, 808, 809, 974, 978  
 \lststyle . . . . . 17, 593, 614, 1338, 1377
- M**
- marvosym (package) . . . . . 145  
 mathdesign (package) . . . . . 16, 104, 158  
 memoir (package) . . . . . 17, 66  
 \Microtype@Hook . . . . . 93, 2904  
 \microtypecontext 12, 16, 1361, 1819,  
     3066, 3288, 3291, 3299, 3302, 3317  
 \microtypesetup . . . . . 8, 11, 2909, 3065  
 minimal (package) . . . . . 39  
 \MT@abbr@ex . . . . . 475  
 \MT@abbr@ex@ec . . . . . 475  
 \MT@abbr@ex@inh . . . . . 475  
 \MT@abbr@kn . . . . . 482  
 \MT@abbr@kn@ec . . . . . 482  
 \MT@abbr@kn@inh . . . . . 482  
 \MT@abbr@nl . . . . . 475  
 \MT@abbr@pr . . . . . 475  
 \MT@abbr@pr@ec . . . . . 475  
 \MT@abbr@pr@inh . . . . . 475  
 \MT@abbr@resp . . . . . 482  
 \MT@abbr@resp@ec . . . . . 482  
 \MT@abbr@resp@inh . . . . . 482  
 \MT@active@features . . . 1779, 1782,  
     1794, 1801, 1836, 1847, 2175,  
     3076, 3173, 3203, 3216, 3273, 3274  
 \MT@addto@setup . . . . . 513, 571, 1738,  
     1819, 1820, 1932, 2043, 2796,  
     2910, 3041, 3201, 3235, 3272, 3279  
 \MT@auto . 1036, 3127, 3140, 3151, 3155  
 \MT@auto@ . . . . . 1024, 1036, 1040, 1098  
 \MT@autofalse . . . . . 125, 3138, 3150  
 \MT@autotruer . . . . . 125, 2876  
 \MT@babel@false . . . . . 132, 3051  
 \MT@babel@true . . . . . 132  
 \MT@begin@catcodes . . . . . 603,  
     1427, 1428, 1451, 2162, 2207,  
     2235, 2254, 2272, 2450, 2889, 2890  
 \MT@catcodes . . . . . 100, 119

- \MT@char ..... 798,  
808, 809, 814, 815, 818, 820,  
835, 840, 865, 974, 975, 977–  
979, 1068, 1080, 1081, 1083,  
1085, 1125, 1127, 1128, 1164,  
1175, 1176, 1182, 1183, 1189,  
1190, 1193, 1195, 1201–1203,  
1205–1208, 1271, 1282, 1283,  
1289, 1290, 1293, 1295, 1301,  
1302, 1304–1306, 1579, 1595,  
1596, 1601, 1671, 1673, 1675,  
1701, 1707, 2507, 2508, 2513, 2516
- \MT@char@ ..... 1579, 1585, 1590,  
1595, 1610, 1612, 1618, 1619,  
1621, 1634, 1635, 1638, 1639,  
1642, 1643, 1647, 1649, 1674, 1715
- \MT@charstring ..... 1592, 1673, 1678
- \MT@check@font ..... 1760, 1776, 1838
- \MT@check@font@cx ..... 1780, 1838
- \MT@check@range ..... 2649, 2651
- \MT@check@range@ ..... 2651, 2652
- \MT@check@rlist ..... 2591, 2642
- \MT@check@rlist@ ..... 2642, 2643
- \MT@checklist@ ..... 654, 667, 1387
- \MT@checklist@family ..... 707
- \MT@checklist@font ..... 686
- \MT@checklist@size ..... 736
- \MT@checksetup .....  
... 2920, 2931, 2938, 2959, 2981
- \MT@clist@break 354, 681, 699, 731, 749
- \MT@clist@function ..... 354
- \MT@cnt@encoding ... 2528, 2535, 2536
- \MT@cnt@family ..... 2533, 2542, 2543
- \MT@cnt@series ..... 2540, 2549, 2550
- \MT@cnt@shape ..... 2547, 2555, 2556
- \MT@config@file .....  
2879, 2887, 2888, 2892, 2893, 2896
- \MT@context ... 1501, 1513, 1532, 1558
- \MT@count ..... 305,  
306, 448, 827, 835, 836, 841,  
842, 847, 848, 852, 860, 897, 959
- \MT@curr@file 1431, 1432, 1439, 1440,  
2164, 2165, 2286, 2471, 2892, 3328
- \MT@curr@list@name .....  
... 858, 900, 1094, 1413,  
1571, 1710, 1718, 1723, 1729, 1734
- \MT@curr@set@name ..... 1900,  
1902–1904, 1906, 1907, 1920,  
1925, 1930, 1933, 1959, 1963,  
1994, 2025, 2031, 2036, 2041,  
2044, 2288, 2290, 2316, 2335, 2472
- \MT@declare@char@inh 2455, 2461, 2469
- \MT@declare@codes ..... 2363, 2375
- \MT@declare@sets 1887, 1893, 1901, 2177
- \MT@DeclareSet ..... 1874, 1875, 1877
- \MT@DeclareSet@ ..... 1878, 1881, 1884
- \MT@DeclareSetAndUseIt 1872, 1873, 1877
- \MT@def@bool@opt .....  
... 2762, 2778, 2782, 2791, 2800
- \MT@def@n ..... 184,  
1866, 1867, 2192, 2211, 2239,  
2258, 2276, 2365, 2421, 2474, 2902
- \MT@def@num@opt ..... 2826, 2841
- \MT@def@simple@bool@opt ... 2778, 2781
- \MT@default@ex@set ..... 2128, 3182
- \MT@default@kn@set ..... 2128, 3224
- \MT@default@pr@set ..... 2128, 3085
- \MT@default@sp@set ..... 2128, 3209
- \MT@define@code@key .....  
... 2298, 2357–2360, 2492–2494
- \MT@define@code@key@font .....  
... 2321, 2362, 2496
- \MT@define@code@key@size .....  
... 2309, 2361, 2495
- \MT@define@context ..... 1843, 1864
- \MT@define@ex@key . 2417, 2430–2432
- \MT@define@key@unit . 2395, 2414, 2415
- \MT@define@option ... 2688, 2724, 2725
- \MT@define@option@ .. 2731, 2759, 2760
- \MT@define@optionX .. 2909, 2969, 2970
- \MT@define@optionX@ . 2972, 3002, 3003
- \MT@define@set@key@ . 1911–1914, 1918
- \MT@define@set@key@font ... 1916, 2029
- \MT@define@set@key@size ... 1915, 1952
- \MT@define@set@keys . 1910, 2080, 2081
- \MT@detokenize@c ..... 257, 1587, 1671
- \MT@detokenize@n ..... 257, 1704, 1705
- \MT@dimen@six ..... 776, 828, 852, 897
- \MT@dinfo ..... 46
- \MT@dinfo@nl ..... 46
- \MT@do@font . 447, 769, 1046, 1213, 1311
- \MT@documentfalse ..... 134
- \MT@documenttrue ..... 134, 3276
- \MT@dofalse .....  
647, 658, 680, 702, 719, 748, 1391
- \MT@dotrue .....  
647, 650, 677, 716, 725, 745, 1383
- \MT@draftfalse ..... 128, 2802
- \MT@drafttrue ..... 128, 2804
- \MT@edef@n .....  
. 185, 1537, 1562, 1859, 2101,  
2105, 2132, 2137, 2148, 2286,  
2304, 2326, 2470, 2483, 2614,  
2622, 2708, 2710, 2745, 2747, 2832
- \MT@encoding ..... 637, 1496, 1508,  
1587, 1671, 1703, 1722, 1728, 1734

- \MT@encoding@check ..... 2481, 2485
- \MT@end@catcodes . 1429, 1468, 2170,  
2212, 2240, 2259, 2277, 2467, 2891
- \MT@error ..... 36, 507, 522
- \MT@ex@cname 1020, 1022, 1099, 1100,  
1117, 1118, 1132, 1539, 2214, 2239
- \MT@ex@context ..... 1830, 1865
- \MT@ex@do ..... 1023, 1058
- \MT@ex@doc@contexts ..... 1865
- \MT@ex@factor ..... 135, 1037, 1103
- \MT@ex@factor@ .....  
1037, 1048, 1051, 1070, 1071, 1098
- \MT@ex@inh@name ..... 1082, 1083, 1085
- \MT@ex@level ..... 135, 3174, 3176
- \MT@ex@max ..... 150, 1073, 1074
- \MT@ex@min ..... 150, 1076, 1077
- \MT@ex@setname .....  
... 2098, 3179, 3180, 3182, 3183
- \MT@ex@split ..... 1060, 1064
- \MT@exp@one@n ..... 190, 360,  
503, 625, 627, 672, 691, 1436,  
1776, 1783, 1815, 1847, 1853,  
1865, 1893, 1932, 2043, 2091,  
2121, 2157, 2461, 3288, 3299, 3317
- \MT@exp@string .....  
189, 622, 779, 867, 874, 1350, 1551
- \MT@exp@two@c ..... 191, 241,  
249, 263, 624, 1586, 1591, 1672
- \MT@exp@two@n .....  
. 192, 712, 721, 1343, 1924, 2035
- \MT@expansion ..... 630, 1014, 3194
- \MT@expansionfalse 124, 3047, 3070, 3168
- \MT@expansiontrue ..... 124, 2875
- \MT@extra@auto ..... 2214, 2437, 2443
- \MT@extra@context .....  
..... 1865, 2196, 2214, 2243,  
2261, 2326, 2330, 2331, 2334,  
2373, 2448, 2588, 2594, 2598,  
2602, 2605, 2610, 2611, 2614, 2648
- \MT@extra@factor .....  
... 2196, 2214, 2243, 2261, 2369
- \MT@extra@load ..... 2196,  
2214, 2243, 2261, 2289, 2290, 2367
- \MT@extra@preset .....  
... 2196, 2214, 2243, 2261, 2371
- \MT@extra@shrink ..... 2214
- \MT@extra@step ..... 2214
- \MT@extra@stretch ..... 2214
- \MT@extra@unit ..... 2196, 2243,  
2261, 2377, 2387, 2398, 2401, 2404
- \MT@factor@default 160, 2844, 2850, 3079
- \MT@family 625, 637, 713, 1497, 2149, 2150
- \MT@familyalias . 626, 627, 643, 720,  
722, 724, 726, 1507, 1509, 2151
- \MT@feat ..... 647, 781, 879,  
881, 883, 884, 886, 887, 893,  
895, 898, 905–909, 911, 913–  
917, 920, 921, 926, 930, 933,  
936, 939, 941, 942, 951, 952,  
957, 1403, 1405, 1408, 1409,  
1413–1415, 1417, 1533, 1535,  
1537, 1541, 1542, 1546, 1550,  
1553, 1560, 1562, 1564, 1565,  
1567, 1570, 1573, 1576, 1711, 1712
- \MT@features ..... 496, 1865, 1887,  
2080, 2085, 2115, 2375, 2455, 2900
- \MT@features@long . 496, 503, 508, 1864
- \MT@file@list ..... 1423, 1425,  
1433, 1436, 1441, 1444, 2157, 2161
- \MT@find@file ..... 625, 627, 1423
- \MT@fix@font@spec ... 1929, 1948, 2040
- \MT@font ..... 622, 624, 695, 698,  
701, 770, 771, 777, 779, 784,  
808, 809, 814, 815, 835, 840,  
855, 860, 865, 867, 874, 965,  
968, 974, 975, 978, 979, 1024,  
1040, 1046, 1080, 1081, 1125,  
1128, 1175, 1176, 1182, 1183,  
1189, 1190, 1201–1203, 1206–  
1208, 1214–1216, 1282, 1283,  
1289, 1290, 1301, 1302, 1305,  
1306, 1312, 1313, 1394, 1480,  
1551, 1737, 1753, 1754, 1756,  
1766, 1776, 1778, 1783, 1805, 1816
- \MT@font@list .. 1736, 1776–1778, 1837
- \MT@get@axis 2051–2054, 2067, 2343–2346
- \MT@get@basefamily ..... 1435, 1469
- \MT@get@char@unit .....  
..... 799, 928, 955, 1165, 1272
- \MT@get@charwd .... 833, 852, 931, 955
- \MT@get@codes@name .....  
... 2201, 2220, 2248, 2266, 2280
- \MT@get@config ..... 2879
- \MT@get@dimen@six 757, 776, 1144, 1251
- \MT@get@ex@opt ..... 1017, 1031, 1098
- \MT@get@ex@opt@ 1105–1107, 1109, 1116
- \MT@get@font ..... 2034, 2047
- \MT@get@font@and@size ..... 2325, 2339
- \MT@get@font@dimen ..... 854, 934
- \MT@get@highlevel .....  
... 1923, 1936, 2069, 2303, 2482
- \MT@get@inh@list .....  
... 760, 1019, 1147, 1254, 1558
- \MT@get@listname .... 1479, 1535, 1560
- \MT@get@listname@ ..... 1479

- \MT@get@opt ..... 758, 904, 1145, 1252  
 \MT@get@range ..... 1956, 1966, 2313  
 \MT@get@size .....  
     1973, 1978, 1989, 2008, 2061, 2354  
 \MT@get@slot ..... 797,  
     1067, 1163, 1270, 1579, 2506, 2512  
 \MT@get@space@unit 928, 1004, 1173,  
     1180, 1187, 1234, 1238, 1280, 1287  
 \MT@get@unit ..... 936, 945  
 \MT@get@unit@ ..... 945  
 \MT@gobble@settings ..... 776  
 \MT@gobblethree ..... 3010, 3014  
 \MT@if@list@exists .....  
     756, 1016, 1030, 1143, 1250, 1532  
 \MT@ifdefined@c@T ..... 196,  
     562, 577, 817, 1082, 1192, 1292,  
     2149, 2222, 2289, 2577, 2904, 2908  
 \MT@ifdefined@c@TF .....  
     ..... 196, 533, 1358, 1536,  
     1561, 2587, 3082, 3179, 3206, 3221  
 \MT@ifdefined@n@T ..... 196, 669,  
     688, 709, 738, 818, 939, 1083,  
     1110, 1193, 1293, 1515, 1903,  
     2144, 2282, 2294, 2536, 2543,  
     2550, 2556, 2588, 2602, 3025, 3315  
 \MT@ifdefined@n@TF .....  
     ..... 196, 643, 652, 670, 689,  
     710, 739, 905, 913, 1099, 1117,  
     1385, 1408, 1503, 1587, 2100,  
     2104, 2130, 2281, 2563, 2621,  
     2637, 2707, 2744, 2901, 3286, 3297  
 \MT@ifdim ..... 314, 420, 421, 425,  
     426, 1991, 2000, 2653–2655,  
     2657, 2658, 2665–2667, 2670, 2671  
 \MT@ifdimen 293, 2020, 2386, 2403, 2864  
 \MT@ifempty 265, 806, 812, 1172, 1179,  
     1186, 1279, 1286, 1372, 1849,  
     1886, 1891, 1938, 1944, 1967,  
     1968, 1981, 1982, 2048, 2055,  
     2070, 2084, 2089, 2114, 2119,  
     2340, 2348, 2365, 2367, 2369,  
     2371, 2373, 2419, 2454, 2459,  
     2486, 2693, 2736, 2764, 2809,  
     2828, 2843, 2854, 2881, 2917, 2978  
 \MT@ifint .. 276, 1641, 2420, 2831, 2846  
 \MT@ifstreq ..... 321,  
     1329, 1404, 1474–1476, 1534,  
     1607, 1814, 2379, 2385, 2400,  
     2402, 2696, 2698, 2701, 2704,  
     2739, 2741, 2767, 2768, 2812,  
     2813, 2818, 2857, 2863, 2919,  
     2926, 2930, 2937, 2980, 2986, 3259  
 \MT@in@clist ..... 380,  
     503, 672, 692, 712, 721, 1343,  
     1425, 1436, 1776, 1783, 1847, 2157  
 \MT@in@rlist ..... 414, 741, 1524  
 \MT@in@rlist@ ..... 414  
 \MT@in@rlist@@ ..... 414  
 \MT@in@tlist ..... 402, 1853  
 \MT@in@tlist@ ..... 402  
 \MT@increment 448, 2535, 2542, 2549, 2555  
 \MT@info 36, 50, 622, 2102, 2922, 2928,  
     2934, 2941, 2983, 2988, 3013, 3017  
 \MT@info@missing@char ... 836, 842, 863  
 \MT@info@nl ..... 36,  
     47, 51, 534, 621, 865, 872, 2808,  
     2888, 3073, 3078, 3083, 3086,  
     3090, 3175, 3180, 3183, 3195,  
     3205, 3207, 3210, 3214, 3220,  
     3222, 3225, 3228, 3240, 3248, 3282  
 \MT@inh@do ..... 1572, 2497  
 \MT@inh@name .. 1570, 1571, 2515, 2522  
 \MT@inh@split ..... 2499, 2503  
 \MT@inlist@false 380, 384, 403, 415, 1791  
 \MT@inlist@true .....  
     ... 380, 386, 410, 422, 427, 1791  
 \MT@is@active ..... 1586, 1654  
 \MT@is@char ..... 1591, 1672, 1678  
 \MT@is@composite ..... 1589, 1699  
 \MT@is@feature .....  
     ..... 502, 1892, 2090, 2120, 2460  
 \MT@is@letter . 1584, 1608, 1675, 1707  
 \MT@is@number ..... 1624, 1629  
 \MT@is@symbol ..... 1588, 1670  
 \MT@iterate ..... 436  
 \MT@kerning ..... 633, 1244, 3251  
 \MT@kerningfalse ..... 131, 3050, 3250  
 \MT@kerningtrue ..... 131, 1354  
 \MT@kn@c@name .....  
     ... 1255, 1257, 1320, 2261, 2276  
 \MT@kn@context ..... 1329, 1831, 1865  
 \MT@kn@do ..... 1258, 1261  
 \MT@kn@doc@contexts ..... 1865  
 \MT@kn@factor ..... 135  
 \MT@kn@factor@ ..... 905, 1330, 1332  
 \MT@kn@inh@name ..... 1292, 1293, 1295  
 \MT@kn@max ..... 150  
 \MT@kn@min ..... 150  
 \MT@kn@setname ..... 1346,  
     1349, 1357, 3221, 3222, 3224, 3225  
 \MT@kn@split ..... 1263, 1267  
 \MT@kn@split@val ..... 1273, 1277  
 \MT@kn@unit ..... 135  
 \MT@kn@unit@ ..... 913, 1323  
 \MT@led@kern ..... 530

- \MT@led@unhbox@line ..... 530  
 \MT@ledmac@setup ..... 530, 573  
 \MT@let@cn ..... 187,  
     188, 644, 763, 1022, 1100, 1150,  
     1257, 1403, 1414, 1529, 1533,  
     1567, 1570, 2012, 2288, 2472, 2829  
 \MT@let@nc ..... 186,  
     942, 1546, 1573, 1576, 1786,  
     1837, 1906, 1920, 2031, 2199,  
     2218, 2229, 2246, 2264, 2290,  
     2623, 2635, 2702, 2705, 2933, 2940  
 \MT@let@nn ..... 188, 906, 911,  
     914, 926, 1118, 1121, 1852, 2295  
 \MT@letterspacing 135, 1359, 3230, 3231  
 \MT@letterspacing@ .....  
     ..... 1331, 1358, 1359, 1371  
 \MT@letterspacing@default .. 164, 3231  
 \MT@listname ..... 1479,  
     1526, 1529, 1536, 1537, 1561, 1562  
 \MT@load@list 761, 1020, 1148, 1255, 1401  
 \MT@loop ..... 436, 443, 2634  
 \MT@lower ..... 1960, 1966, 2316  
 \MT@make@string . 474, 1950, 2143, 2156  
 \MT@map@clist@ ..... 354  
 \MT@map@clist@c ..... 354, 1782,  
     1794, 1801, 1836, 1864, 1887,  
     1889, 2080, 2085, 2087, 2115,  
     2117, 2375, 2455, 2457, 2900, 3023  
 \MT@map@clist@n .... 354, 651, 1384,  
     1865, 1921, 1954, 2032, 2190,  
     2199, 2202, 2217, 2221, 2246,  
     2249, 2264, 2267, 2301, 2311,  
     2323, 2509, 2691, 2734, 2914, 2975  
 \MT@map@tlist@ ..... 370  
 \MT@map@tlist@c .....  
     .... 370, 405, 416, 819, 1084,  
     1194, 1294, 1483, 1807, 2589, 2647  
 \MT@map@tlist@n . 370, 1463, 2779, 2839  
 \MT@maybe@do 647, 754, 1014, 1138, 1245  
 \MT@MT 35, 36, 37, 40, 42, 43, 95, 523,  
     525, 2882, 3013, 3017, 3074, 3263  
 \MT@next@listname ..... 1486, 1495  
 \MT@next@listname@ .. 1504, 1516, 1522  
 \MT@nl@setname ..... 2178  
 \MT@noligatures ..... 635, 1380  
 \MT@noligaturesfalse ..... 127  
 \MT@noligaturestrue ..... 127, 2176  
 \MT@nonselectedfalse ..... 1027, 1041  
 \MT@nonselectedtrue ..... 1027, 1029  
 \MT@noresetfalse .....  
     ... 1606, 1607, 1614, 1623, 1644  
 \MT@noresettrue ..... 1582, 1606, 1632  
 \MT@old@cmd ..... 26, 31–34  
 \MT@opt@autofalse ..... 2686  
 \MT@opt@autotruetrue ..... 2686  
 \MT@opt@DVIfalse ..... 2687  
 \MT@opt@DVIttrue ..... 2687, 2784, 2787  
 \MT@opt@expansionfalse ..... 2685  
 \MT@opt@expansiontrue ..... 2685  
 \MT@orig@add@accent ..... 1769  
 \MT@orig@foreign@language . 3294, 3296  
 \MT@orig@pickupfont .. 591, 1742, 1771  
 \MT@orig@select@language .. 3283, 3285  
 \MT@pdfcprot@error ..... 521, 572  
 \MT@pdftex@no ... 53, 87, 91, 178, 3154  
 \MT@permute .....  
     2206, 2234, 2253, 2271, 2477, 2527  
 \MT@permute@ ..... 2527  
 \MT@permute@@ ..... 2527  
 \MT@permute@@@ ..... 2527  
 \MT@permute@@@@ ..... 2527  
 \MT@permute@@@@@ ..... 2554, 2559  
 \MT@permute@@@@@@ ..... 2567, 2569  
 \MT@permute@define .....  
     ..... 2560, 2570–2572, 2619  
 \MT@permute@reset ..... 2530, 2625  
 \MT@permute@reset@ .. 2626–2629, 2632  
 \MT@permutelist .....  
     ..... 2196, 2232, 2251, 2269,  
     2326, 2327, 2334, 2475, 2583,  
     2584, 2588, 2594, 2598, 2602–  
     2605, 2614, 2615, 2648, 2679, 2680  
 \MT@pickupfont ..... 1768, 1773  
 \MT@pr@c@name 761, 763, 984, 2196, 2211  
 \MT@pr@context ..... 1830, 1865  
 \MT@pr@do ..... 764, 788  
 \MT@pr@doc@contexts ..... 1865  
 \MT@pr@factor .. 135, 2847, 3079, 3080  
 \MT@pr@factor@ ..... 905  
 \MT@pr@inh@name ..... 817, 818, 820  
 \MT@pr@level ..... 135, 3077, 3078  
 \MT@pr@max ..... 150  
 \MT@pr@min ..... 150  
 \MT@pr@setname .....  
     ... 2098, 3082, 3083, 3085, 3086  
 \MT@pr@split ..... 790, 794  
 \MT@pr@split@val ..... 800, 804  
 \MT@pr@unit ..... 135, 2865, 3081  
 \MT@pr@unit@ ..... 913, 987  
 \MT@preset@aux ..... 989, 991, 993,  
     994, 997, 1325, 1327, 1334, 1335  
 \MT@preset@aux@factor .....  
     ..... 989, 997, 1228–1230, 1325  
 \MT@preset@aux@space .....  
     ..... 991, 997, 1232, 1327  
 \MT@preset@ex ..... 1112, 1131

- \MT@preset@kn ..... 1318
- \MT@preset@kn@ ..... 1318
- \MT@preset@pr ..... 982
- \MT@preset@pr@ ..... 982
- \MT@preset@sp ..... 1221
- \MT@preset@sp@ ..... 1221
- \MT@ProcessOptionsWithKV .. 3020, 3040
- \MT@protrusion ..... 629, 754, 3089
- \MT@protrusionfalse ..... 123, 3046
- \MT@protrusiontrue ..... 123, 2872
- \MT@rbba@expansion ..... 490
- \MT@rbba@kerning ..... 490
- \MT@rbba@protrusion ..... 490
- \MT@rbba@spacing ..... 490
- \MT@register@font ... 1763, 1778, 1839
- \MT@register@font@cx ..... 1800, 1839
- \MT@register@subst@font .....  
..... 1757, 1777, 1840
- \MT@register@subst@font@cx 1793, 1840
- \MT@rem@from@clist ..... 391, 1815
- \MT@rem@from@lists ..... 1809, 1813
- \MT@repeat ..... 436, 445, 2640
- \MT@requires@etex .....  
165, 196, 257, 449, 458, 833, 863
- \MT@requires@pdftex ..... 177,  
276, 293, 321, 532, 825, 1049,  
1137, 1244, 1338, 1380, 2173,  
2436, 2442, 2874, 3129, 3200, 3254
- \MT@reset@context ..... 1825, 1828
- \MT@reset@ef@codes .....  
..... 1018, 1039, 1049, 1113
- \MT@reset@ef@codes@ .. 1044, 1052, 1056
- \MT@reset@kn@codes .. 1253, 1259, 1309
- \MT@reset@kn@codes@ ..... 1309
- \MT@reset@pr@codes ..... 759, 765, 774
- \MT@reset@pr@codes@ ..... 774
- \MT@reset@sp@codes .. 1146, 1152, 1211
- \MT@reset@sp@codes@ ..... 1211
- \MT@restore@catcodes ..... 112, 120
- \MT@restore@p@h 559, 567, 568, 589, 599
- \MT@saved@setupfont ..... 3011, 3018
- \MT@scale .....  
458, 828, 880, 894, 897, 1071, 1331
- \MT@scale@factor 830, 849, 878, 999, 1133
- \MT@scale@to@em .....  
807, 813, 825, 846, 1005, 1174,  
1181, 1188, 1235, 1239, 1281, 1288
- \MT@selectedfalse ..... 126
- \MT@selectedtrue ..... 126
- \MT@series ..... 637, 1498, 1510
- \MT@set@all@ex ..... 1044, 1134
- \MT@set@all@kn ..... 1309, 1336
- \MT@set@all@pr ..... 767, 774, 995
- \MT@set@all@sp ..... 1211, 1242
- \MT@set@default@set . 2115, 2121, 2128
- \MT@set@ex@codes .... 1043, 3159, 3161
- \MT@set@ex@codes@n .. 1027, 1043, 3161
- \MT@set@ex@codes@s ..... 1015, 3159
- \MT@set@ex@heirs ..... 1086, 1124
- \MT@set@ex@list ..... 2236, 2238
- \MT@set@inh@list ..... 2451, 2453
- \MT@set@kn@codes ..... 1249
- \MT@set@kn@heirs ..... 1296, 1300
- \MT@set@kn@list ..... 2273, 2275
- \MT@set@listname ..... 762,  
940, 1021, 1111, 1149, 1256, 1710
- \MT@set@opt 2202, 2221, 2249, 2267, 2293
- \MT@set@pr@codes ..... 755
- \MT@set@pr@heirs ..... 821, 973
- \MT@set@pr@list ..... 2208, 2210
- \MT@set@sp@codes ..... 1142
- \MT@set@sp@heirs ..... 1196, 1200
- \MT@set@sp@list ..... 2255, 2257
- \MT@setup@ 512, 515, 516, 519, 2794, 2795
- \MT@setup@contexts ..... 1822, 1835
- \MT@setupfont .....  
620, 1762, 3011, 3014, 3018, 3053
- \MT@setupfont@hook ..... 560,  
574, 576, 581, 589, 599, 608, 623
- \MT@shape ..... 637, 1499, 1511
- \MT@shrink .....  
. 135, 1034, 3096, 3098, 3100,  
3104, 3105, 3108, 3112, 3164, 3177
- \MT@shrink@ ... 1024, 1034, 1040, 1098
- \MT@shrink@default ..... 161, 3100
- \MT@size .. 421, 425, 426, 637, 744, 747
- \MT@size@name ..... 414, 1526
- \MT@sp@c@name .....  
... 1148, 1150, 1223, 2243, 2258
- \MT@sp@context ..... 1831, 1865, 3259
- \MT@sp@do ..... 1151, 1154
- \MT@sp@doc@contexts ..... 1865
- \MT@sp@factor ..... 135
- \MT@sp@factor@ ..... 905
- \MT@sp@inh@name ..... 1192, 1193, 1195
- \MT@sp@max ..... 150
- \MT@sp@min ..... 150
- \MT@sp@setname 3206, 3207, 3209, 3210
- \MT@sp@split ..... 1156, 1160
- \MT@sp@split@val ..... 1166, 1170
- \MT@sp@unit ..... 135
- \MT@sp@unit@ ..... 913, 1226
- \MT@spacing .... 632, 1137, 3213, 3243
- \MT@spacingfalse ..... 130, 3049, 3242
- \MT@spacingtrue ..... 130
- \MT@split@name ..... 624, 637

- `\MT@split@val` ..... 1170  
`\MT@step` .....  
     135, 1035, 3103, 3119, 3126, 3178  
`\MT@step@` ..... 1024, 1035, 1040, 1098  
`\MT@step@default` ..... 161  
`\MT@stretch` ..... 135,  
     1033, 3093, 3094, 3097, 3098,  
     3104, 3106, 3111, 3114, 3163, 3177  
`\MT@stretch@` .. 1024, 1033, 1040, 1098  
`\MT@stretch@default` ..... 161, 3094  
`\MT@temp` ..... 2050, 2078, 2342  
`\MT@tempencoding` 2562, 2563, 2573, 2580  
`\MT@tempfamily` ..... 2574  
`\MT@tempseries` ..... 2575  
`\MT@tempshape` ..... 2576  
`\MT@tempsize` 2577, 2587, 2590, 2595, 2630  
`\MT@test@ast` ..... 1937, 1942  
`\MT@testrest` ..... 1606  
`\MT@tlist@break` 370, 411, 433, 1488, 2682  
`\MT@toks` ..... 122, 515,  
     516, 865, 873, 900, 1094, 1583,  
     1612, 1619, 1664, 1715, 1727, 1733  
`\MT@try@order` ..... 1483, 1491  
`\MT@undefined@char` .. 1659, 1661, 1669  
`\MT@upper` ..... 1960, 1966, 2316  
`\MT@use@set` ..... 2085, 2091, 2098  
`\MT@val` ..... 1849–  
     1851, 1854, 1856, 1859, 1922,  
     1926, 1937, 1939, 1955–1957,  
     1969, 1972, 1974, 1977, 1979,  
     1980, 1988, 1990, 1991, 1993,  
     1996, 1998, 2009, 2012, 2016,  
     2020–2022, 2024, 2026, 2033,  
     2034, 2037, 2057, 2059, 2062–  
     2064, 2068, 2070, 2072, 2074,  
     2302, 2304, 2312–2314, 2324,  
     2325, 2329, 2335, 2350, 2352,  
     2480, 2481, 2483, 2487, 2508,  
     2515, 2521, 2522, 2692, 2693,  
     2696, 2698, 2701, 2704, 2707,  
     2708, 2713, 2735, 2736, 2739,  
     2741, 2744, 2745, 2750, 2915,  
     2917, 2919, 2926, 2930, 2937,  
     2945, 2976, 2978, 2980, 2986, 2991  
`\MT@vinfo` .... 36, 47, 621, 664, 908,  
     917, 921, 957, 1101, 1119, 1395,  
     1409, 1432, 1440, 1443, 1539,  
     1541, 1548, 1829, 1851, 2159,  
     2165, 2808, 2817, 3287, 3298, 3316  
`\MT@warn@ascii` ..... 1621, 1714  
`\MT@warn@axis@empty` .....  
     ..... 2056, 2071, 2076, 2349  
`\MT@warn@code@too@large` . 884, 887, 891  
`\MT@warn@err` ..... 36, 2814, 2815  
`\MT@warn@ex@too@large` 1074, 1077, 1092  
`\MT@warn@number@too@large` . 1648, 1720  
`\MT@warn@preset@towidth` .....  
     ..... 988, 1008, 1227, 1324  
`\MT@warn@rest` ..... 1600, 1725  
`\MT@warn@unknown` ..... 1597, 1731  
`\MT@warning` 27, 36, 1365, 1405, 1417,  
     1550, 1904, 1992, 2024, 2077,  
     2107, 2134, 2145, 2167, 2182,  
     2283, 2380, 2389, 2406, 2423,  
     2439, 2488, 2581, 2603, 2679,  
     2814, 2858, 2905, 2944, 2963, 2990  
`\MT@warning@nl` ... 36, 90, 546, 551,  
     583, 778, 856, 898, 950, 1009,  
     1093, 1715, 1721, 1726, 1732,  
     1744, 2224, 2712, 2749, 2769,  
     2815, 2819, 2833, 2848, 2867,  
     2895, 3043, 3121, 3132, 3144,  
     3165, 3237, 3245, 3260, 3312, 3321  
`\MT@while@num` .... 441, 447, 604–606  
`\MT@with@package` .... 520, 529, 564,  
     568, 572, 573, 579, 598, 602, 613  
`\MT@xadd` ..... 340,  
     1348, 1433, 1441, 1444, 1795,  
     1803, 1856, 1958, 2161, 2315, 2514  
`\MT@xaddb` ..... 347, 2333, 2593
- N**
- `ngerman (package)` ..... 22  
`\normalfont` ..... 619, 3187, 3191  
`\nullfont` ..... 619
- P**
- `\pdfadjustinterwordglue` ... 3002, 3204  
`\pdfadjustspacing` ... 2970, 3174, 3190  
`\pdfappendkern` ..... 1355, 3004, 3219  
`pdfcprot (package)` .....  
     ..... 5, 18, 19, 37, 44, 66, 154  
`\pdffontexpand` ..... 1024, 1040  
`\pdfmatch` ..... 278, 295  
`\pdfnoligatures` ..... 1394  
`\pdfoutput` ..... 57, 2784, 2785,  
     2787, 2788, 2873, 3068, 3073, 3130  
`\pdfprependkern` .... 1356, 3003, 3218  
`\pdfprotrudechars` ..... 2969, 3077  
`\pdfstrcmp` ..... 323  
`\pdftexrevision` ..... 56, 61, 70, 72  
`\pdftexversion` .....  
     ..... 55, 59–61, 63, 65, 67, 69, 77  
`\pickup@font` ..... 591,  
     1743, 1745, 1750, 1768, 1771, 1773  
`pifont (package)` ..... 66

- protrusion (option) . . . . 5, 6, 10, 88, 89  
 PSNFSS (package) . . . . . 149  
 pstricks (package) . . . . . 7, 97  
 pxfonts (package) . . . . . 16, 104
- Q**
- qfonts (package) . . . . . 16, 104, 158
- R**
- relsize (package) . . . . . 73  
 \rightmarginkern . . . . . 539  
 \rPCODE . . . . . 771, 814, 815, 975, 979
- S**
- selected (option) . . . . . 7,  
     13, 14, 22, 49, 60, 90, 155, 156  
 \selectfont . . . . . 1824, 1833, 3327  
 \set@fontsize . . . . . 2016  
 \SetExpansion . . . . . 7, 13, 2214, 3057  
 \SetExtraKerning . . . . . 15, 2261, 3060  
 \SetExtraSpacing . . . . . 14, 2243, 3059  
 \SetProtrusion . . . . . 6, 11, 2196, 3056  
 \sfcode . . . . . 3258  
 \shbcode . 1189, 1190, 1203, 1208, 1216  
 \showhyphens . . . . . 3185, 3189  
 shrink (option) . . 7, 14, 22, 91, 98, 158  
 slantsc (package) . . . . . 129
- soul (package) . . . . . 39, 150, 159  
 spacing (option) . . . . . 89  
 \stbcode . 1182, 1183, 1202, 1207, 1215  
 step (option) . . . 7, 22, 91, 98, 155, 158  
 stretch (option) . . . 7, 14, 18, 22, 91, 98
- T**
- tex4ht (package) . . . . . 21, 38, 159  
 \textls . . . . . 18, 594, 615, 1371  
 tipa (package) . . . . . 66  
 \tracingmicrotype . . . . . 46  
 txfonts (package) . . . . . 16, 104
- U**
- ucs (package) . . . . . 18, 63  
 unit (option) . . . . . 6, 21, 157, 158  
 \UseMicrotypeSet . . . . .  
     . . 3, 10, 22, 32, 1882, 2082, 3055
- V**
- verbose (option) . . . . . 8, 22, 91, 156
- X**
- xkeyval (package) . . . . . 81
- Z**
- zefonts (package) . . . . . 16, 104