



# 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)

v1.9f — 2006/09/09

## 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 requires pdf $\TeX$  1.30. The package will by default enable the features that can safely be assumed to work.

This work may be distributed and/or modified under the conditions of the [L<sup>A</sup>T<sub>E</sub>X Project Public License](#), either version 1.3c of this license or (at your option) any later version. This work has the LPPL maintenance status ‘author-maintained’.

---

<sup>1</sup> 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, URW Garamond, Adobe Garamond and Minion, Bitstream Charter, and the AMS symbols and Euler fonts, for various Euro symbol fonts, as well as some generic settings for unknown fonts. Contributions are very welcome.

## Contents

<b>1</b>	<b>Micro-Typography with pdf<math>\TeX</math></b>	<b>4</b>
<b>2</b>	<b>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>20</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 . . . . .	35
13.4	Setting up a font . . . . .	38
	[Protrusion – 42] [Expansion – 48] [Interword Space (Glue) – 51]	
	[Additional Kerning – 53] [Letterspacing – 55] [Disabling	
	Ligatures – 55] [Loading the Configuration – 56] [Translating Characters	
	into Slots – 60] [Hook into $\LaTeX$ 's font selection – 65] [Context-sensitive	
	Setup – 68]	
13.5	Configuration . . . . .	70
	[Font Sets – 70] [Interaction with babel – 77] [Fine Tuning – 77] [Character	
	Inheritance – 82] [Permutation – 84]	

13.6 Package Options . . . . .	87
[Declaring the Options – 87] [Reading the Configuration File – 91] [Hook for Other Packages – 92] [Changing Options Later – 93] [Processing the Options – 95]	
<b>14 Configuration Files</b>	<b>101</b>
14.1 Font Sets . . . . .	101
14.2 Font Aliases . . . . .	102
14.3 Interaction with babel . . . . .	103
14.4 Note on Admissible Characters . . . . .	103
14.5 Character Inheritance . . . . .	104
[OT1 – 104] [T1 – 104] [LY1 – 105] [OT4 – 105] [QX – 106] [T5 – 106] [Euro symbols – 107]	
14.6 Font Expansion . . . . .	108
14.7 Character Protrusion . . . . .	109
[Default – 109] [Italics – 116] [Small Caps – 124] [Italic Small Caps – 127] [textcomp – 129] [Computer Modern math – 133] [AMS symbols – 137] [Euler – 141] [Euro symbols – 145]	
14.8 Interword Spacing . . . . .	146
[Nonfrenchspacing – 147]	
14.9 Additional Kerning . . . . .	148
[French – 149] [Turkish – 149] [Letterspacing – 149]	
<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>158</b>
<b>C The L<sup>A</sup>T<sub>E</sub>X Project Public License</b>	<b>164</b>

## List of Tables

1 Predefined font sets . . . . .	10
2 Fonts with tailored protrusion settings . . . . .	16
3 Order for matching font attributes . . . . .	58

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

In the summer of 2004, Hàn Thế Thành implemented a feature that has proven as a major facilitation for 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. The microtype package 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 percent signs before '\betatrue' 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). In other words, microtype will try to apply as much micro-typography as can safely be expected to work under the respective conditions (and it is usually not necessary to load the package with different options for PDF resp. DVI mode).

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

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

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

```
\usepackage{microtype}
```

<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, hence the default value is `nocompatibility`.

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

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.

6 Modern TeX systems are using pdfTeX as the default engine even for DVI output.

<b>draft</b>	true, false	false
<b>final</b>	If the draft option is passed to the package, <i>all micro-typographic extensions will be disabled</i> . The draft and final options may also be inherited from the class options; of course, you can override them in the package options.	
<b>verbose</b>	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.	
<b>config</b>	<i>&lt;file name&gt;</i>	microtype
	Various settings for this package will be loaded from a main configuration file, by default microtype.cfg (see section 5.4). You can have a different configuration file loaded instead by specifying its name <i>without the extension</i> , e.g., config=mycrotpe.	

### 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 incompatibility (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 ‘A’ are valid, provided the character is actually declared in both the input and the font encoding. You also have the possibility to declare lists of characters that should inherit protrusion or expansion factors (see section 5.3).

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, just with a different factor applied to them:

```
\SetProtrusion
[ factor = 700
  load   = cmr-T1 ]
{ encoding = T1,
  family   = cmr,
  size     = large- }
{ }
```

**unit** By default, the protrusion factors are relative to the respective character's width. The unit option may be used to override this and make microtype regard all values in the list as thousandths of the specified width. Issuing, for instance, 'unit=1em' would have the effect that a value of, say, 50 now results in the character being protruded by 5% of an em of the font (thus simulating the internal measuring of pdfTeX's \lcode and \rcode primitives). The default behaviour can be restored with unit=character.<sup>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.

**inputenc** Select an input encoding that should apply to this list, regardless of what the document's input encoding is. You may specify any encoding that can be loaded via the inputenc package, e.g., ansinew, koi8-r, utf8.

<sup>7</sup> For the interested, table 3 on page 58 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.

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

## 5.2 Font Expansion

**\SetExpansion** [*options*] {*set of fonts*} {*expansion settings*}

By default, all characters of a font are allowed to be stretched or shrunk by the same amount. However, it is also possible to limit the expansion of certain characters if they are more sensitive to deformation. This is the purpose of the `\SetExpansion` command. Note that it will only have an effect if the package has been loaded with the selected option. Otherwise, the expansion settings will be ignored.

*The expansion settings* consist of  $\langle \text{character} \rangle = \langle \text{expansion factor} \rangle$  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, inputenc, context** Analogous to `\SetProtrusion`, the optional argument may be used to assign a name to the list, to load another list, to preset all expansion factors, to set the input encoding, or to determine the context of the list.

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

As a practical example, suppose you have a paragraph containing a widow that could easily be avoided by shrinking the font a little bit more. You could take advantage of the stretch and shrink options to allow for more expansion in this particular paragraph. There is one problem that has to be worked around, however: pdf<sub>T</sub><sub>E</sub>X prohibits the use of the same font with different expansion parameters. If you do not want to create a clone of the font setup (this would require duplicating the `tfm/vf` files under a new name, and writing new `fd` files and map entries), you could exploit a dirty trick and load a minimally larger font for the paragraph in question. E. g., for a document 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,
  step = 5 ]
{ encoding = *,
  size = 10.001 }
{ }
\newcommand{\expandpar}[1]{\%
  \fontsize{10.001}{\baselineskip}\selectfont #1\par}}
% ...
\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 pdf<sub>T</sub>E<sub>X</sub>: 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 À, Á, Â, Ã, Ä, Å and Æ should probably be protruded by the same (absolute) amount as the character A. Using the command `\DeclareCharacterInheritance`, you may declare such classes of characters, so that you then only have to set up the respective base character. With the optional argument, which may contain a comma-separated

list of features, you can confine the scope of the list. Additionally, it accepts the `inputenc` key to set the input encoding for this list. The font set can be declared in the usual way, with the only exception that exactly one encoding must be specified. The inheritance lists are declared as pairs of *⟨base character⟩ = ⟨list of inheriting characters⟩*. Unless you are using a different encoding or a very peculiarly shaped font, there should be no need to change the default character inheritance settings.

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

## 5.4 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 symbols and Euler fonts and Euro symbol 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 (cmsy, cmm)	OML/OMS	n/it
AMS symbols (msa, msb)	U	n
Euler (eur, eus, euf) <sup>h</sup>	U	n
Euro symbols (Adobe, ITC, marvosym)	U/OT1	n, it

<sup>a</sup> Incomplete

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

<sup>c</sup> 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)

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

`\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.

<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.



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

```
\SetProtrusion
[ context = 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 pdf<sub>T</sub><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.

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

*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:<sup>12</sup>

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

<sup>12</sup> If you are using the `fancyvrb` or the `listings` package, this is not necessary, since their implementation of the corresponding environments will inhibit protrusion anyway.

*Compatibility.* The package should work happily together with all other L<sup>A</sup>T<sub>E</sub>X packages (except pdfcprot). 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 – you should then specify the inputenc key.
- The CJK package, like microtype, hooks into the L<sup>A</sup>T<sub>E</sub>X font selection scheme. Therefore, both packages probably don't cooperate well. However, since I know nothing about CJK, I would appreciate feedback on the interaction of both packages – be it positive or negative.

*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 T<sub>E</sub>X system is not set up to embed (or 'download') the base PostScript fonts (e. g., Times, Helvetica, Courier). In most T<sub>E</sub>X distributions, this can be changed in the file updmap.cfg by setting pdfTeXDownloadBase14 to true. Otherwise, consult the local guide of your T<sub>E</sub>X system.
- Warning: pdflatex (file ecrm1000+20): Font ecrm1000+20 at 1200 not found  
Furthermore, automatic font expansion requires Type 1 fonts. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. This is not (yet) possible, unless you manually create expanded instances of the fonts.
- ! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.  
Such an error message could occur if you are trying to employ font expansion while creating DVI output. Remember, that *automatic* font expansion only works when running pdfT<sub>E</sub>X in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your T<sub>E</sub>X system.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf\_mem\_size)=65536].  
When applying micro-typographic enhancement to a large document with a lot of fonts, pdfT<sub>E</sub>X may be running out of memory. The memory can be increased by setting pdf\_mem\_size to a larger value (maximum 524 288). For teT<sub>E</sub>X-based systems, change the settings in texmf.cnf, for MiK<sub>T</sub>E<sub>X</sub>, in the file miktex.ini. Beginning with version 1.30 of pdfT<sub>E</sub>X, memory will grow dynamically, so that this problem can no longer 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>13</sup>

## 10 Acknowledgments

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

*Harald Harders* has contributed protrusion settings for Adobe Minion. I would also like to thank him for a number of bug reports and suggestions he had to make. *Andreas Böhmann* has suggested the possibility to specify ranges of font sizes, and resourcefully assisted in implementing this. He also came up with some good ideas for the management of complex configurations. 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  $\TeX$  typesetting system*, Diss. Masaryk University Brno 2000, in: *TUGBoat*, vol. 21(2000), no. 4, pp. 317–434. (Online at <http://www.tug.org/TUGboat/Articles/tb21-4/tb69thanh.pdf>)

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

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

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, *The pdfTeX user manual*, 16 February 2006. (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>X 2<sub>ε</sub> font selection*, 27 November 2005. (Available from CTAN at [/macros/latex/doc/fntguide.pdf](#))

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

## 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.

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

- New key ‘inputenc’ to specify the lists’ input encodings
- Protrusion settings for Euler math fonts

### 1.9d (2006/05/05)

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

### 1.9c (2006/02/02)

- Protrusion settings for URW Garamond

### 1.9a (2005/12/05)

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

### 1.9 (2005/10/28)

- New command \DisableLigatures to disable ligatures of fonts (requires pdfTeX version 1.30 or later; see section 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)
- ‘DVInoutput’ option will work with T<sub>E</sub>XLive 2004

- 1.8 (2005/06/23)
- If font substitution has occurred, the settings for the substitute will be used instead of those for the selected font
  - New command `\DeclareMicrotypeSetDefault` to declare the default font sets (see section 4)
  - New option ‘`config`’ to load a different configuration file (see section 3.4)
  - 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 (2005/03/23)
- Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings (see sections 4 and 5)
  - Always take font size into account when trying to find protrusion resp. expansion settings for a given font (see section 5)
  - New command `\LoadMicrotypeFile` to load a font configuration file manually (see section 5.4)
  - 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
- 1.6 (2005/01/24)
- New option ‘`factor`’ to influence protrusion resp. expansion of all characters of a font or font set (see sections 3.2 and 5)
  - When pdf $\TeX$  is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled (see section 3.1)
  - Protrusion settings of digits improved
  - Use e- $\TeX$  extensions, if available
- 1.5 (2004/12/15)
- When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled (see section 3.1)
  - New option ‘`selected`’ to enable selected expansion (see sections 3.3 and 5.2); default is: `false`
  - New default for expansion option ‘`step`’:  $4 \cdot (\min(\text{stretch}, \text{shrink})/5)$  (see section 3.3)
  - Protrusion settings for Bitstream Charter
- 1.4b (2004/11/26)
- `\UseMicrotypeSet` requires the set to be declared (see section 4)

- 1.4 (2004/11/12)
  - Set up fonts independently from L<sup>A</sup>T<sub>E</sub>X 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’
- 1.2 (2004/10/03)
  - New font sets: ‘allmath’ and ‘basicmath’ (see section 4 and table 1)
  - Protrusion settings for Computer Modern Roman math symbols
  - Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond
  - If an alias font name is specified, it will be used as an alternative, not as a replacement (see section 5.4)
  - More tests for sanity of settings and whether all fonts will be set up
  - More robust parsing of sizes in font sets
- 1.1 (2004/09/21)
  - 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 (2004/09/11)
  - 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 pdfTeX.

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 Euler Fraktur font (mt-euf.cfg).

eur: Settings for Euler Roman font (mt-eur.cfg).

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

cfg-e: Surrounds Euro symbol configurations.

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

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

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

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

And now for something completely different.

1 *<package>*

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

```
2 \newcommand*\DeclareMicrotypeSet[3] [] {}
3 \newcommand*\UseMicrotypeSet[2] [] {}
4 \newcommand*\DeclareMicrotypeSetDefault[2] [] {}
5 \newcommand*\DeclareMicrotypeAlias[2] {}
6 \newcommand*\SetProtrusion[3] [] {}
```



```

7 \newcommand*\SetExpansion[3] [] {}
8 \newcommand*\DisableLigatures[1] {}
9 \newcommand*\DeclareCharacterInheritance[3] [] {}
10 \newcommand*\LoadMicrotypeFile[1] {}
11 \newcommand*\microtypesetup[1] {}
12 \newcommand*\microtypecontext[1] {}
13 < *beta >
14 \newcommand*\SetExtraSpacing[3] [] {}
15 \newcommand*\SetExtraKerning[3] [] {}
16 \newcommand*\DeclareMicrotypeBabelHook[2] {}
17 \newcommand*\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\DeclareMicroTypeAlias\DeclareMicrotypeAlias
32 \MT@old@cmd\DeclareMicroTypeSet \DeclareMicrotypeSet
33 \MT@old@cmd\UseMicroTypeSet \UseMicrotypeSet
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@n1 40 This error message appears because you loaded the ``\MT@MT'\MessageBreak
\MT@vinfo 41 package with the option `verbose=errors'. Consult the documentation\MessageBreak
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 pdf<sub>T</sub><sub>E</sub>X's features for which we provide an interface here haven't always been available, and some specifics have changed over time. Therefore, we have to test which pdf<sub>T</sub><sub>E</sub>X we're using, if any. \MT@pdftex@no will be used throughout the package to respectively do the right thing.

Currently, there are six cases for pdf<sub>T</sub><sub>E</sub>X:

- 0: not running pdf<sub>T</sub><sub>E</sub>X
- 1: pdf<sub>T</sub><sub>E</sub>X (< 0.14f)
- 2: + micro-typographic extensions (0.14f, 0.14g)
- 3: + protrusion relative to 1em (≥ 0.14h)
- 4: + automatic font expansion; default \efcode = 1000 (≥ 1.20)
- 5: + \left, right marginkern; \pdfnoligatures; \pdfstrcmp (≥ 1.30)
- 6: + adjustment of interword spacing; extra kerning; \pdfmatch<sup>14</sup> (≥ 1.40)

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

A hack circumventing the T<sub>E</sub>XLive 2004 hack which undefines the pdf<sub>T</sub><sub>E</sub>X primitives in the format in order to hide the fact that pdf<sub>T</sub><sub>E</sub>X is being run from the user. This has been *fixed* in T<sub>E</sub>XLive 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           \def\MT@pdftex@no{3}
69           \ifnum\pdftexversion = 14
70             \ifnum\expandafter\pdftexrevision < `h
71               \def\MT@pdftex@no{2}
72             \ifnum\expandafter\pdftexrevision < `f

```

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

```

73         \def\MT@pdftex@no{1}
74         \fi
75     \fi
76 \else
77     \ifnum\pdftexversion < 14
78         \def\MT@pdftex@no{1}
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 pdfTeX 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@if@ A scratch if.
123 \newif\ifMT@if@

\ifMT@protrusion These are the global switches ...
\ifMT@expansion 124 \newif\ifMT@protrusion
\ifMT@auto 125 \newif\ifMT@expansion
\ifMT@selected 126 \newif\ifMT@auto
\ifMT@noligatures 127 \newif\ifMT@selected
\ifMT@draft 128 \newif\ifMT@noligatures
129 \newif\ifMT@draft
130 \beta
\ifMT@spacing 131 \newif\ifMT@spacing
\ifMT@kerning 132 \newif\ifMT@kerning
\ifMT@babel 133 \newif\ifMT@babel
134 \beta

\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
\MT@sp@factor 144 \let\MT@sp@factor\@m
145 \let\MT@kn@factor\@m
\MT@sp@unit
\MT@kn@factor Default unit for spacing settings is space, default unit for kerning is 1em.
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 pdfTeX.
\MT@pr@max 150 \def\MT@pr@min{-\@m}
\MT@ex@min 151 \let\MT@pr@max\@m
\MT@ex@max 152 \let\MT@ex@min\z@
153 \let\MT@ex@max\@m
\MT@sp@min 154 \beta
\MT@sp@max 155 \def\MT@sp@min{-\@m}
156 \let\MT@sp@max\@m
\MT@kn@min 157 \def\MT@kn@min{-\@m}
\MT@kn@max 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 161 \def\MT@stretch@default{20 }
\MT@step@default 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 }

```

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

165 `\newif\ifMT@document`

## 13.2 Auxiliary macros

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

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

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

```
178 \def\MT@requires@pdftex#1{%
179   \ifnum\MT@pdftex@no<#1\relax
180     \expandafter\@secondoftwo
181   \else
182     \expandafter\@firstoftwo
183   \fi
184 }
```

`\MT@gllet` The forgotten primitive.

185 `\def\MT@gllet{\global\let}`

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

```
\MT@gdef@n 186 \def\MT@def@n#1{\expandafter\def\csname #1\endcsname}
187 \def\MT@gdef@n#1{\expandafter\gdef\csname #1\endcsname}
```

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

```
\MT@xdef@n 188 \def\MT@edef@n#1{\expandafter\edef\csname #1\endcsname}
189 \def\MT@xdef@n#1{\expandafter\xdef\csname #1\endcsname}
```

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

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

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

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

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

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

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

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

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

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

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

195 `\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.

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

```

197 \expandafter\expandafter\expandafter
198   #1\expandafter\expandafter\expandafter
199   {\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 200 \MT@requires@etex{
\MT@ifdefined@c@TF 201   \def\MT@ifdefined@c@T#1{\ifdefined#1%
\MT@ifdefined@n@T 202     \expandafter\@firstofone\else\expandafter\@gobble\fi
203   }
204   \def\MT@ifdefined@c@TF#1{\ifdefined#1%
205     \expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
206   }
207   \def\MT@ifdefined@n@T#1{\ifcsname#1\endcsname
208     \expandafter\@firstofone\else\expandafter\@gobble\fi
209   }
210   \def\MT@ifdefined@n@TF#1{\ifcsname#1\endcsname
211     \expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
212   }
213 }{
214   \def\MT@ifdefined@c@T#1{\ifx#1\@undefined
215     \expandafter\@gobble\else\expandafter\@firstofone\fi
216   }
217   \def\MT@ifdefined@c@TF#1{\ifx#1\@undefined
218     \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
219   }
220   \def\MT@ifdefined@n@T#1{\begingroup\MT@exp@two@c\endgroup
221     \ifx\csname #1\endcsname\relax
222       \expandafter\@gobble\else\expandafter\@firstofone\fi
223   }
224   \def\MT@ifdefined@n@TF#1{\begingroup\MT@exp@two@c\endgroup
225     \ifx\csname #1\endcsname\relax
226       \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
227   }
228 }

```

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

`\MT@detokenize@c`

```

229 \MT@requires@etex{
230   \def\MT@detokenize@n#1{\detokenize\expandafter{\string#1}}
231   \def\MT@detokenize@c#1{\detokenize
232     \expandafter\expandafter\expandafter{\expandafter\string#1}}
233 }{
234   \def\MT@detokenize@n#1{\string#1}
235   \def\MT@detokenize@c#1{\MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty}
236 }

```

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

```

237 \begingroup
238 \catcode`\%=12
239 \catcode`\&=14
240 \gdef\MT@ifempty#1{&
241   \if %#1%&
242     \expandafter\@firstoftwo
243   \else
244     \expandafter\@secondoftwo
245   \fi
246 }

```

```

247 \endgroup

\MT@ifint Test whether argument is an integer, using an old trick by Mr. Arseneau, or the
latest and greatest from pdfTeX.
248 \MT@requires@pdftex6{
249   \def\MT@ifint#1{%
250     \ifcase\pdfmatch{^[0-9]+ *$}{#1}\relax
251     \expandafter\@secondoftwo
252   \else
253     \expandafter\@firstoftwo
254   \fi
255 }
256 }{
257   \def\MT@ifint#1{%
258     \if!\ifnum9<1#1!\else?\fi
259     \expandafter\@firstoftwo
260   \else
261     \expandafter\@secondoftwo
262   \fi
263 }
264 }

\MT@ifdimen Test whether argument is dimension (or number). (nd and nc are new Didot resp.
Cicero, added in pdfTeX 1.30; px is a pixel, it seems.)
265 \MT@requires@pdftex6{
266   \def\MT@ifdimen#1{%
267     \ifcase\pdfmatch{^([0-9]+([.][0-9]+)?|[.][0-9]+)%
268               (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
269     \expandafter\@secondoftwo
270   \else
271     \expandafter\@firstoftwo
272   \fi
273 }
274 }{
275   \def\MT@ifdimen#1{%
276     \setbox\z@=\hbox{%
277       \MT@count=1#1\relax
278       \ifnum\MT@count=\@ne
279         \aftergroup\@secondoftwo
280       \else
281         \aftergroup\@firstoftwo
282       \fi
283     }%
284   }
285 }

\MT@ifdim Test floating point numbers.
286 \def\MT@ifdim#1#2#3{%
287   \ifdim #1\p@ #2 #3\p@
288     \expandafter\@firstoftwo
289   \else
290     \expandafter\@secondoftwo
291   \fi
292 }

\MT@ifstreq Test whether two strings (fully expanded) are equal.
293 \MT@requires@pdftex5{
294   \def\MT@ifstreq#1#2{%
295     \ifcase\pdfstrcmp{#1}{#2}\relax
296     \expandafter\@firstoftwo
297   \else
298     \expandafter\@secondoftwo

```

```

299   \fi
300 }
301 }{
302   \def\MT@ifstreq#1#2{%
303     \edef\x{#1}%
304     \edef\y{#2}%
305     \ifx\x\y
306       \expandafter\@firstoftwo
307     \else
308       \expandafter\@secondoftwo
309     \fi
310   }
311 }

\MT@xadd Add item to a list.
312 \def\MT@xadd#1#2{%
313   \ifx#1\relax
314     \xdef#1{#2}%
315   \else
316     \xdef#1{#1#2}%
317   \fi
318 }

\MT@xaddb Add item to the beginning.
319 \def\MT@xaddb#1#2{%
320   \ifx#1\relax
321     \xdef#1{#2}%
322   \else
323     \xdef#1{#2#1}%
324   \fi
325 }

\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@ 326 \def\MT@map@clist@n#1#2{%
\MT@clist@function 327   \ifx\@empty#1\else
\MT@clist@break 328     \def\MT@clist@function##1{#2}%
329     \MT@map@clist@#1,\@nil,\@nnil
330   \fi
331 }
332 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
333 \def\MT@map@clist@#1,{%
334   \ifx\@nil#1%
335     \expandafter\MT@clist@break
336   \fi
337   \MT@clist@function{#1}%
338   \MT@map@clist@
339 }
340 \let\MT@clist@function\@gobble
341 \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@ 342 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break 343 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
344 \def\MT@map@tlist@#1#2{%
345   \ifx\@nnil#2\else
346     #1{#2}%
347     \expandafter\MT@map@tlist@
348     \expandafter#1%
349   \fi
350 }

```



```

351 \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 352 \newif\ifMT@inlist@
353 \def\MT@in@clist#1#2{%
354   \def\x##1,#1,##2##3\@nnil{%
355     \ifx##2\@empty
356       \MT@inlist@false
357     \else
358       \MT@inlist@true
359     \fi
360   }%
361   \expandafter\x\expandafter,#2,#1,\@empty\@nnil
362 }

\MT@rem@from@clist Remove item <#1> from comma list <#2>. This is basically \@removeelement from
ltnctr1.dtx. Using \pdfmatch and \pdflastmatch here would be really slow!
363 \def\MT@rem@from@clist#1#2{%
364   \def\x##1,#1,##2\x{##1,##2\y}%
365   \def\y##1,\y##2\y{\ifx,##1\@empty\else##1\fi}%
366   \xdef#2{\MT@exp@two@c\y\x\expandafter,#2,\y,#1,\x}%
367 }

\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.
368 \def\MT@in@tlist#1#2{%
369   \MT@inlist@false
370   \def\x{#1}%
371   \MT@map@tlist@c#2\MT@in@tlist@
372 }
373 \def\MT@in@tlist@#1{%
374   \edef\y{#1}%
375   \ifx\x\y
376     \MT@inlist@true
377     \expandafter\MT@tlist@break
378   \fi
379 }

\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@@ 380 \def\MT@in@rlist#1{%
\MT@size@name 381   \MT@inlist@false
382   \MT@map@tlist@c#1\MT@in@rlist@
383 }
384 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
385 \def\MT@in@rlist@@#1#2#3{%
386   \MT@ifdim{#2}=\mone{%
387     \MT@ifdim{#1}=\MT@size
388     \MT@inlist@true
389     \relax
390   }{%
391     \MT@ifdim\MT@size<{#1}\relax{%
392       \MT@ifdim\MT@size<{#2}%
393       \MT@inlist@true
394       \relax
395     }%
396   }%
397   \ifMT@inlist@
398     \def\MT@size@name{#3}%
399     \expandafter\MT@tlist@break

```

```

400 \fi
401 }

\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 402 \def\MT@loop#1\MT@repeat{%
403 \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
404 \MT@iterate \let\MT@iterate\relax
405 }
406 \let\MT@repeat\fi

\MT@while@num Execute <#3> from <#1> up to (excluding) <#2>.
407 \def\MT@while@num#1#2#3{%
408 \@tempcnta#1\relax
409 \MT@loop #3%
410 \advance\@tempcnta \@ne
411 \ifnum\@tempcnta < #2\MT@repeat
412 }

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

\MT@count Increment macro <#1> by one. Saves using up too many counters. The e-TEX way is
\MT@increment slightly faster.
414 \newcount\MT@count
415 \MT@requires@etex{
416 \def\MT@increment#1{\edef#1{\number\numexpr #1 + 1\relax}}
417 }{
418 \def\MT@increment#1{%
419 \MT@count=#1\relax
420 \advance\MT@count \@ne
421 \edef#1{\number\MT@count}%
422 }
423 }

\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.
424 \MT@requires@etex{
425 \def\MT@scale#1#2#3{%
426 \ifnum #3 = \z@
427 #1=\numexpr #1 * #2\relax
428 \else
429 #1=\numexpr #1 * #2 / #3\relax
430 \fi
431 }
432 }{
433 \def\MT@scale#1#2#3{%
434 \multiply #1 #2\relax
435 \ifnum #3 = \z@ \else
436 \divide #1 #3\relax
437 \fi
438 }
439 }

\MT@make@string Set the category code of all characters to 12.
440 \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
\MT@abbr@pr@c
\MT@abbr@ex@c
\MT@abbr@pr@inh
\MT@abbr@ex@inh
\MT@abbr@n1

```

output.

```
441 \def\MT@abbr@pr{protrusion}
442 \def\MT@abbr@ex{expansion}
443 \def\MT@abbr@pr@c{protrusion codes}
444 \def\MT@abbr@ex@c{expansion codes}
445 \def\MT@abbr@pr@inh{protrusion inheritance}
446 \def\MT@abbr@ex@inh{expansion inheritance}
447 \def\MT@abbr@nl{noligatures}
```

\MT@abbr@sp Also for the experimental features.

```
\MT@abbr@sp@c 448 <beta>
449 \def\MT@abbr@sp{spacing}
\MT@abbr@sp@inh 450 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@kn 451 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@kn@c 452 \def\MT@abbr@kn{ Kerning}
\MT@abbr@kn@inh 453 \def\MT@abbr@kn@c{ Kerning codes}
454 \def\MT@abbr@kn@inh{ Kerning inheritance}
455 </beta>
```

\MT@rbba@protrusion These we also need the other way round.

```
\MT@rbba@expansion 456 \def\MT@rbba@protrusion{pr}
\MT@rbba@spacing 457 \def\MT@rbba@expansion{ex}
458 <beta>
\MT@rbba@kerning 459 \def\MT@rbba@spacing{sp}
460 \def\MT@rbba@kerning{kn}
461 </beta>
```

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

```
\MT@features@long 462 \def\MT@features{pr,ex%
463 <beta> ,sp,kn%
464 }
465 \def\MT@features@long{protrusion,expansion%
466 <beta> ,spacing,kerning%
467 }
```

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

```
468 \def\MT@is@feature#1{%
469 \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
470 \ifMT@inlist@
471 \expandafter\@firstofone
472 \else
473 \MT@error{`\@tempa' is not an available micro-typographic\MessageBreak
474 feature. Ignoring #1}{Available features are: ` \MT@features@long'.}%
475 \expandafter\@gobble
476 \fi
477 }
```

### 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.

```
478 \let\MT@setup@\@empty
```

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

```
479 \def\MT@addto@setup{\g@addto@macro\MT@setup@}
```

It will be executed at the end of the preamble, and emptied (the combine class will call it repeatedly).

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

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

```
481 \def\MT@with@package#1{\ifpackageloaded{#1}\@firstofone\@gobble}
```

`\MT@pdfcp@error` Our competitor, the pdfcp package, must not be tolerated!

```
482 \def\MT@pdfcp@error{%
483   \MT@error{Detected the 'pdfcp' package!\MessageBreak
484     '\MT@MT' and 'pdfcp' may not be used together}{%
485     The 'pdfcp' package provides an interface to character protrusion.\MessageBreak
486     So does the '\MT@MT' package. Using both packages at the same\MessageBreak
487     time will almost certainly lead to undesired results. Have your choice!}%
488   \let\MT@pdfcp@error\relax
489 }
490 \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 off the lines one by one. This will destroy character protrusion. (There aren't any problems with the lineno package, since it takes a different approach.) — ... —

`\MT@led@unhbox@line` 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).

```
491 \MT@requires@pdftex5{
492   \def\MT@ledmac@setup{%
493     \ifMT@protrusion
494       \MT@ifdefined@c@TF\l@dunhbox@line{%
495         \MT@info@n1{Patching ledmac to enable character protrusion}%
496         \newdimen\MT@led@kern
497         \let\MT@led@unhbox@line\l@dunhbox@line
498         \renewcommand*{\l@dunhbox@line}[1]{%
499           \ifhbox##1%
500             \MT@led@kern=\rightmarginkern##1%
501             \kern\leftmarginkern##1%
502             \MT@led@unhbox@line##1%
503             \kern\MT@led@kern
504           \fi
505         }%
506       }%
507     \MT@warning@n1{%
508       Character protrusion in paragraphs with line\MessageBreak
509       numbering will only work if you update ledmac}%
510   }%
511   \fi
512 }
513 }
```

```

514 \def\MT@ledmac@setup{%
515   \ifMT@protrusion
516     \MT@warning@nl{%
517       The pdftex version you are using does not allow\MessageBreak
518       character protrusion in paragraphs with line\MessageBreak
519       numbering by the `ledmac' package.\MessageBreak
520       Upgrade pdftex to version 1.30 or later}%
521   \fi
522 }
523 }

```

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

```

524 \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`.)

```

525 \def\MT@setupfont@hook{%

```

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

```

526 \@ifpackagewith{babel}{spanish}{%
527   \MT@ifdefined@c@T\percentsign{\let\%\percentsign}%
528 } \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.)

```

529 \MT@with@package{csquotes}{%
530   \@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax
531 }%

```

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

```

532 \@ifpackageloaded{hyperref}\MT@restore@p@h{%
533   \MT@with@package{tex4ht}\MT@restore@p@h
534 }%
535 }

```

Check again at the end of the preamble.

```

536 \MT@addto@setup{%
537   \MT@with@package{pdfcpot}\MT@pdfcpot@error
538   \MT@with@package{ledmac}\MT@ledmac@setup

```

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

```

539 \let\MT@setupfont@hook\@empty
540 \@ifpackagewith{babel}{spanish}{%
541   \g@addto@macro\MT@setupfont@hook{%
542     \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
543 } \relax
544 \MT@with@package{csquotes}{%
545   \@ifpackagelater{csquotes}{2005/05/11}{%
546     \g@addto@macro\MT@setupfont@hook\@disablequotes
547   }%
548   \MT@warning@nl{%

```

```

549         Should you receive warnings about unknown slot\MessageBreak
550         numbers, try upgrading the `csquotes' package}%
551     }%
552 }%
553 \ifpackageloaded{hyperref}{%
554     \g@addto@macro\MT@setupfont@hook\MT@restore@p@h

```

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

```

555     \pdfstringdefDisableCommands{%
556         \let\pickup@font\MT@orig@pickupfont
557     }%
558     \let\lststyle\empty
559     \def\textls#1#{\@firstofone}%
560 }%
561 }%
562 }%
563 \MT@with@package{tex4ht}{%
564     \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
565 }%
566 }%

```

The `listings` package makes numbers and letters active.

```

567 \MT@with@package{listings}{%
568     \g@addto@macro\MT@begin@catcodes{%
569         \MT@while@num{"30}{\catcode\@tempcnta 12\relax}%
570         \MT@while@num{"41}{\catcode\@tempcnta 11\relax}%
571         \MT@while@num{"61}{\catcode\@tempcnta 11\relax}%
572     }%

```

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.

```

573     \g@addto@macro\MT@setupfont@hook{%
574         \let\lst@ProcessLetter\empty
575     }%
576 }%

```

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.

```

577 }%
578 \MT@with@package{soul}{%
579     \soulregister\lststyle 0%
580     \soulregister\textls 1%
581 }%
582 }%
583 }

```

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

```

584 \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`), and if so, adjusting `\lcode` and `\rcode` (protrusion) and `\efcode` (expansion) for each character.

```

585 \def\MT@setupfont{%
586     \ifx\MT@vinfo\MT@info@nl

```

```
587 \MT@info{Setting up font \MT@font'}\fi
```

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

```
588 \MT@setupfont@hook
```

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

```
589 \MT@exp@two@c\MT@split@name\string\MT@font\@nil
```

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

```
590 \MT@exp@one@n\MT@find@file\MT@family
591 \ifx\MT@familyalias\@empty \else
592 \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.)

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

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

Protrusion has to be set up first, says Thành!

```
594 \MT@protrusion
595 \MT@expansion
```

Interword spacing and kerning.

```
596 <*\beta>
597 \MT@spacing
598 \MT@kerning
599 </\beta>
```

Disable ligatures?

```
600 \MT@noligatures
601 }
```

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

```
\MT@encoding 602 \def\MT@split@name#1/#2/#3/#4/#5\@nil{%
\MT@family 603 \def\MT@encoding{#1}%
\MT@series 604 \def\MT@family{#2}%
\MT@shape 605 \def\MT@series{#3}%
\MT@size 606 \def\MT@shape{#4}%
\MT@size 607 \def\MT@size{#5}%
```

\MT@familyalias Alias family?

```
608 \MT@ifdefined@n@TF{MT@MT@family @alias}%
609 {\MT@let@cn\MT@familyalias{MT@MT@family @alias}}%
610 {\let\MT@familyalias\@empty}%
611 }
```

\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 612 \newif\ifMT@do
613 \def\MT@maybe@do#1{%
```

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

```
614 \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).

```

615 \MT@dotrue
616 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
617 \MT@ifdefined@n@TF{MT@checklist@##1}%
618 {\csname MT@checklist@##1\endcsname}%
619 {\MT@checklist@{##1}}%
620 {#1}%
621 }%
622 \else
623 \MT@dofalse
624 \fi
625 \ifMT@do

```

\MT@feat stores the current feature.

```

626 \def\MT@feat{#1}%
627 \csname MT@set@#1@codes\endcsname
628 \else
629 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
630 \fi
631 }

```

\MT@checklist@ The generic test.

```

632 \def\MT@checklist@#1#2{%
633 \edef\@tempa{\csname MT@#2@setname\endcsname}%
634 !debug \MT@ifdefined@n@T
635 debug \MT@ifdefined@n@TF
636 {MT@#2list@#1@\@tempa}{%

```

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

```

637 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
638 \csname MT@#1\expandafter\endcsname
639 \csname MT@#2list@#1@\@tempa\endcsname
640 \ifMT@inlist@
641 debug \MT@dinfo@n1{1}{\@nameuse{MT@abbr@#2}: #1 ~\@nameuse{MT@#1}' in list}%
642 \MT@dotrue
643 \else
644 debug \MT@dinfo@n1{1}{\@nameuse{MT@abbr@#2}: #1 ~\@nameuse{MT@#1}' not in list}%
645 \MT@dofalse
646 \expandafter\MT@clist@break
647 \fi
648 }%

```

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.

```

649 debug {\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#2}: #1 list empty}}%
650 }

```

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

```

651 \def\MT@checklist@font#1{%
652 \edef\@tempa{\csname MT@#1@setname\endcsname}%
653 !debug \MT@ifdefined@n@T
654 debug \MT@ifdefined@n@TF
655 {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.

```

656 \expandafter\expandafter\expandafter\MT@exp@one@n
657 \expandafter\expandafter\expandafter\MT@in@clist
658 \expandafter\expandafter\expandafter{%
659 \expandafter\expandafter\expandafter\string

```



```

660      \expandafter\MT@font\expandafter}%
661      \csname MT@#1list@font@\@tempa\endcsname
662      \ifMT@inlist@
663 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: font `\'MT@@font' in list}%
664      \expandafter\MT@clist@break
665      \else
666 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: font `\'MT@@font' not in list}%
667      \MT@dofalse
668      \fi
669    }%
670 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: font list empty}}%
671 }

```

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

```

672 \def\MT@checklist@family#1{%
673   \edef\@tempa{\csname MT@#1@setname\endcsname}%
674 <!debug> \MT@ifdefined@n@T
675 <debug> \MT@ifdefined@n@TF
676   {MT@#1list@family@\@tempa}%
677   \MT@exp@two@n\MT@in@clist
678   \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
679   \ifMT@inlist@
680 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family `\'@nameuse{MT@family}' in list}%
681   \MT@dotrue
682   \else
683 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family `\'@nameuse{MT@family}' not in list}%
684   \MT@dofalse
685   \ifx\MT@familyalias\@empty \else
686     \MT@exp@two@n\MT@in@clist
687     \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
688     \ifMT@inlist@
689 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: alias `\'MT@familyalias' in list}%
690     \MT@dotrue
691 <debug> \else\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: alias `\'MT@familyalias' not in list}%
692     \fi
693     \fi
694     \fi
695     \ifMT@do \else
696       \expandafter\MT@clist@break
697     \fi
698   }%
699 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: family list empty}}%
700 }

```

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

```

701 \def\MT@checklist@size#1{%
702   \edef\@tempa{\csname MT@#1@setname\endcsname}%
703 <!debug> \MT@ifdefined@n@T
704 <debug> \MT@ifdefined@n@TF
705   {MT@#1list@size@\@tempa}%
706   \expandafter\MT@in@rlist
707   \csname MT@#1list@size@\@tempa\endcsname
708   \ifMT@inlist@
709 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size `\'MT@size' in list}%
710   \MT@dotrue
711   \else
712 <debug> \MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size `\'MT@size' not in list}%
713   \MT@dofalse
714   \expandafter\MT@clist@break
715   \fi
716 }%
717 <debug> {\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: size list empty}}%

```

```
718 }
```

### 13.4.1 Protrusion

`\MT@protrusion` Set up for protrusion?

```
719 \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.

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

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

```
721 \MT@if@list@exists{%
```

```
722 \MT@get@dimen@six
```

```
723 \MT@get@opt
```

```
724 \MT@reset@pr@codes
```

Get the name of the inheritance list and parse it.

```
725 \MT@get@inh@list
```

Set an input encoding?

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

Load additional lists?

```
727 \MT@load@list\MT@pr@c@name
```

```
728 \MT@set@listname
```

Load the main list.

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

```
730 \expandafter\MT@pr@do\@tempc,\relax,%
```

```
731 }\MT@reset@pr@codes
```

```
732 }
```

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

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

```
734 <debug>\MT@info@nl{3}{-- lp/rp: setting all to \number#1/\number#2}%
```

```
735 \MT@do@font{%
```

```
736 \lcode\MT@font\@tempcnta=#1\relax
```

```
737 \rcode\MT@font\@tempcnta=#2\relax
```

```
738 }%
```

```
739 }
```

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

`\MT@reset@pr@codes@`

```
740 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
```

```
741 \let\MT@reset@pr@codes\relax
```

`\MT@gobble@settings` If `\fontdimen6` is zero, character protrusion won't work, and we can skip the settings (for example, the dsfont fonts don't specify this dimension; this is probably a bug).

`\MT@dimen@six`

`\MT@get@dimen@six`

```
742 \def\MT@get@dimen@six{%
```

```
743 \ifnum\fontdimen6\MT@font=\z@
```

```
744 \MT@warning@nl{%
```

```
745 Font '\MT@font' does not specify its\MessageBreak
```

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

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

```
748 \expandafter\MT@gobble@settings
```

```
749 \else
```

```

750 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
751 \fi
752 }
753 \def\MT@gobble@settings#1\@tempc,\relax,{}

```

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

```

754 \def\MT@pr@do#1,{%
755 \ifx\relax#1\@empty\else
756 \MT@pr@split #1=\relax
757 \expandafter\MT@pr@do
758 \fi
759 }

```

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

```

760 \def\MT@pr@split#1=#2=#3\relax{%
761 \def\@tempa{#1}%
762 \ifx\@tempa\@empty \else
763 \MT@get@slot
764 \ifnum\MT@char > \m@ne
765 \MT@get@char@unit
766 \MT@pr@split@val#2\relax
767 \fi
768 \fi
769 }

```

\MT@pr@split@val

```

770 \def\MT@pr@split@val#1,#2\relax{%
771 \def\@tempb{#1}%
772 \MT@ifempty\@tempb\relax{%
773 \MT@scale@to@em
774 \lcode\MT@font\MT@char=\@tempcntb
775 (debug)\MT@info@n{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char: [#1]}%
776 }%
777 \def\@tempb{#2}%
778 \MT@ifempty\@tempb\relax{%
779 \MT@scale@to@em
780 \rcode\MT@font\MT@char=\@tempcntb
781 (debug)\MT@info@n{4}{;;; rp (\MT@char): \number\rpcode\MT@font\MT@char: [#2]}%
782 }%

```

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

```

783 \MT@ifdefined@cT\MT@pr@inh@name{%
784 \MT@ifdefined@nT\MT@inh@\MT@pr@inh@name @\MT@char @}{%
785 \expandafter\MT@map@tlist@c
786 \csname MT@inh@\MT@pr@inh@name @\MT@char @\endcsname
787 \MT@set@pr@heirs
788 }%
789 }%
790 }

```

\MT@scale@to@em Since pdf<sub>TEX</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 \lcode resp. \rcode, since this would disallow protrusion factors larger than the character width (since \[lr]pcode’s limit is 1000). Now, the maximum protrusion is 1em of the font.

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

```
791 \MT@requires@pdftex3{
792   \def\MT@scale@to@em{%
793     \@tempcntb=\MT@count\relax
```

For really huge fonts (100pt or so), an arithmetic overflow could occur with vanilla  $\TeX$ . Using  $\epsilon\text{-}\TeX$ , this can't happen, since the intermediate value is 64 bit, which could only be reached with a character width larger than `\maxdimen`.

```
794   \MT@scale\@tempcntb \@tempb \MT@dimen@six
795   \ifnum\@tempcntb=\z@ \else
796     \MT@scale@factor
797   \fi
798 }
```

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

```
799 \MT@requires@etex{
800   \def\MT@get@charwd{%
801     \MT@count=\fontcharwd\MT@font\MT@char\relax
802     \ifnum\MT@count=\z@ \MT@info@missing@char \fi
803   }
804   {}{
805     \def\MT@get@charwd{%
806       \setbox\z@=\hbox{\MT@font \char\MT@char}%
807       \MT@count=\wd\z@\relax
808       \ifnum\MT@count=\z@ \MT@info@missing@char \fi
809     }
810   }
```

No adjustment with versions 0.14f and 0.14g.

```
811 }}{
812   \def\MT@scale@to@em{%
813     \MT@count=\@tempb\relax
814     \ifnum\MT@count=\z@ \else
815       \MT@scale@factor
816     \fi
817   }
```

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

```
818   \def\MT@get@charwd{\MT@count=\MT@dimen@six}
819 }
```

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

```
820 \def\MT@get@font@dimen#1{%
821   \ifnum\fontdimen#1\MT@font=\z@
822     \MT@warning@nl{Font \font@name does not specify its\MessageBreak
823       \backslashchar fontdimen \number#1 (it's zero).\MessageBreak
824       You should use a different 'unit' for \MT@curr@list@name}%
825   \else
826     \MT@count=\fontdimen#1\MT@font
827   \fi
828 }
```

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

```
829 \MT@requires@etex{
830   \def\MT@info@missing@char{%
831     \MT@info@nl{Character '\the\MT@toks' \iffontchar\MT@font\MT@char
832       has a width of 0pt \else is missing \fi in font\MessageBreak
833       '\MT@font'. Ignoring protrusion settings\MessageBreak
834       for this character}%
```

```

835 }
836 }{
837 \def\MT@info@missing@char{%
838 \MT@info@n1{%
839 Character '\the\MT@toks' has a width of 0pt\MessageBreak
840 (it's probably missing) in font '\MT@font'.\MessageBreak
841 Ignoring protrusion settings for this character}%
842 }
843 }

```

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

```

844 \def\MT@scale@factor{%
845 \ifnum\csname MT@\MT@feat @factor@endcsname=\@m \else
846 \expandafter\MT@scale\expandafter
847 \@tempcntb \csname MT@\MT@feat @factor@endcsname \@m
848 \fi
849 \ifnum\@tempcntb>\csname MT@\MT@feat @max@endcsname\relax
850 \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @max@endcsname
851 \else
852 \ifnum\@tempcntb<\csname MT@\MT@feat @min@endcsname\relax
853 \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @min@endcsname
854 \fi
855 \fi
856 }

```

`\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.

```

857 \def\MT@warn@code@too@large#1{%
858 \@tempcnta=#1\relax
859 \ifnum\csname MT@\MT@feat @factor@endcsname=\@m \else
860 \expandafter\MT@scale\expandafter\@tempcnta\expandafter\@m
861 \csname MT@\MT@feat @factor@endcsname
862 \fi
863 \MT@scale\@tempcnta \MT@dimen@six \MT@count
864 \MT@warning@n1{The \nameuse{MT@abbr@\MT@feat} code \@tempb\space
865 is too large for character\MessageBreak
866 '\the\MT@toks' in \MT@currlist@name.\MessageBreak
867 Setting it to the maximum of \number\@tempcnta}%
868 \@tempcntb=#1\relax
869 }

```

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

```

870 \def\MT@get@opt{%
871 \MT@set@listname

```

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

```

\MT@sp@factor@ 872 \MT@ifdefined@nTF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name@endcsname @factor}{%
\MT@kn@factor@ 873 \MT@let@nn{MT@\MT@feat @factor@}
874 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name@endcsname @factor}%
875 \MT@vinfo{... : Multiplying \nameuse{MT@abbr@\MT@feat} codes by
876 \number\csname MT@\MT@feat @factor@endcsname/1000}%
877 }{%
878 \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
879 }%

```

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

```

\MT@kn@unit@ 880 \MT@ifdefined@nTF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name@endcsname @unit}{%
881 \MT@let@nn{MT@\MT@feat @unit}%

```

```

882      {MT@MT@feat @c@\csname MT@MT@feat @c@name\endcsname @unit}%
883 \expandafter\ifx\csname MT@MT@feat @unit\endcsname\empty
884   \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
885     relative to character widths}%
886 \else
887   \expandafter\ifx\csname MT@MT@feat @unit\endcsname\m@ne
888   \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
889     relative to width of space}%
890 \fi
891 \fi
892 }{%
893   \MT@let@nn{MT@MT@feat @unit@}{MT@MT@feat @unit}%
894 }%

```

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

\MT@get@char@unit

```

895 \let\MT@get@char@unit\relax
896 \let\MT@get@space@unit@gobble
897 \expandafter\ifx\csname MT@MT@feat @unit\endcsname\empty
898   \let\MT@get@char@unit\MT@get@charwd
899 \else
900   \expandafter\ifx\csname MT@MT@feat @unit\endcsname\m@ne
901     \let\MT@get@space@unit\MT@get@font@dimen
902   \else
903     \expandafter\MT@get@unit\csname MT@MT@feat @unit\endcsname
904   \fi
905 \fi

```

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

```

906 \MT@ifdefined@nT{MT@MT@feat @c@\csname MT@MT@feat @c@name\endcsname @preset}{%
907   \csname MT@preset@MT@feat\endcsname
908   \MT@let@nc{MT@reset@MT@feat @codes}\relax
909 }%
910 }

```

\MT@get@unit

\MT@get@unit@

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

```

911 \def\MT@get@unit#1{%
912   \expandafter\MT@get@unit@#1 e!\@nil
913   \ifx\x\@empty\else\let#1\x\fi
914   \@defaultunits\@tempdima#1 pt\relax\@nnil
915   \ifdim\@tempdima=\z@
916     \MT@warning@n1{%
917       Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
918       width. Setting factors of list ~\@nameuse{MT@MT@feat @c@name}'\MessageBreak
919       relative to character widths instead}%
920     \let#1\@empty
921     \let\MT@get@char@unit\MT@get@charwd
922   \else
923     \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
924       to \the\@tempdima}%
925     \MT@count=\@tempdima\relax
926   \fi
927 }
928 \def\MT@get@unit@#1e#2#3\@nil{%
929   \ifx\#3\\\let\x\@empty \else
930     \if m#2%
931       \edef\x{#1\fontdimen6\MT@font}%

```

```

932 \else
933 \if x#2%
934 \edef\x{#1\fontdimen5\MT@font}%
935 \fi
936 \fi
937 \fi
938 }

```

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

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

```

939 \def\MT@set@inputenc#1{%
940 \def\MT@cat{#1}%
941 \MT@ifdefined@nT{%
942 \MT@MT@feat @#1@csname MT@MT@feat @#1@name\endcsname @inputenc}{%
943 \MT@ifdefined@c@TF\inputencoding{%
944 <debug>\MT@dinfo@n1{1}{input encoding: \@nameuse{%
945 <debug>\MT@MT@feat @#1@csname MT@MT@feat @#1@name\endcsname @inputenc}}%
946 \inputencoding{\@nameuse{%
947 \MT@MT@feat @#1@csname MT@MT@feat @#1@name\endcsname @inputenc}}}%
948 }{%
949 \MT@error{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
950 \MessageBreak package isn't loaded. Ignoring key}%
951 {You must load the `inputenc' package before you can use the `inputenc' key.}%
952 }%
953 }%
954 }

```

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

```

955 \def\MT@set@pr@heirs#1{%
956 \lpcode\MT@font#1=\lpcode\MT@font\MT@char
957 \rprcode\MT@font#1=\rprcode\MT@font\MT@char
958 <*debug>
959 \MT@dinfo@n2{2}{-- heir of \MT@char: #1}%
960 \MT@dinfo@n4{4}{;;; lp/rp (#1): \number\lpcode\MT@font\MT@char/%
961 \number\rprcode\MT@font\MT@char}%
962 </debug>
963 }

```

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

```

\MT@preset@pr@
964 \def\MT@preset@pr{%
965 \expandafter\expandafter\expandafter\MT@preset@pr@
966 \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
967 }
968 \def\MT@preset@pr@#1,#2\@nil{%
969 \ifx\MT@pr@unit@\@empty
970 \MT@warn@preset@twidth{pr}%
971 \let\MT@preset@aux\MT@preset@aux@factor
972 \else
973 \let\MT@preset@aux\MT@preset@aux@space
974 \fi
975 \MT@preset@aux{#1}\@tempa
976 \MT@preset@aux{#2}\@tempb
977 \MT@set@all@pr\@tempa\@tempb
978 }

```

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

```

\MT@preset@aux@factor 979 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 980 \@tempcntb=#1\relax
981 \MT@scale@factor
982 \edef#2{\number\@tempcntb}%
983 }
984 \def\MT@preset@aux@space#1#2{%

```

```

985 \def\@tempb{#1}%
986 \MT@get@space@unit\tw@
987 \MT@scale@to@em
988 \edef#2{\number\@tempcntb}%
989 }

```

\MT@warn@preset@tewidth

```

990 \def\MT@warn@preset@tewidth#1{%
991 \MT@warning@n1{%
992 Cannot preset characters relative to their widths\MessageBreak
993 for \@nameuse{MT@abbr@#1} list "\@nameuse{MT@#1@cc@name}'. Presetting them%
994 \MessageBreak relative to 1em instead}%
995 }

```

### 13.4.2 Expansion

\MT@expansion Set up for expansion?

```

996 \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).

```

997 \def\MT@set@ex@codes@s{%
998 \MT@if@list@exists{%
999 \MT@get@ex@opt
1000 \MT@reset@ef@codes
1001 \MT@get@inh@list
1002 \MT@set@inputenc{c}%
1003 \MT@load@list\MT@ex@cc@name
1004 \MT@set@listname
1005 \MT@let@cn\@tempc{MT@ex@cc\MT@ex@cc@name}%
1006 \expandafter\MT@ex@do\@tempc,\relax,%
1007 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1008 }\relax
1009 }

```

\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.

```

1010 \newif\ifMT@nonselected
1011 \def\MT@set@ex@codes@n{%
1012 \MT@nonselectedtrue
1013 \MT@if@list@exists
1014 \MT@get@ex@opt
1015 {%
1016 \let\MT@stretch@\MT@stretch
1017 \let\MT@shrink@\MT@shrink
1018 \let\MT@step@\MT@step
1019 \let\MT@auto@\MT@auto
1020 \let\MT@ex@factor@\MT@ex@factor
1021 }%
1022 \MT@reset@ef@codes
1023 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1024 \MT@nonselectedfalse
1025 }

```



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

```
1026 \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  
`\MT@reset@ef@codes@` factor of this font).

```
1027 \def\MT@set@all@ex#1{%
1028 <debug>\MT@info@n1{3}{-- ex: setting all to \number#1}%
1029 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1030 }
1031 \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.

```
1032 \MT@requires@pdfTeX4{
1033 \def\MT@reset@ef@codes{%
1034 \ifnum\MT@ex@factor@=\@m \else
1035 \MT@reset@ef@codes@
1036 \fi
1037 }
1038 }{
1039 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1040 }
```

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

```
1041 \def\MT@ex@do#1,{%
1042 \ifx\relax#1\@empty \else
1043 \MT@ex@split #1=\relax
1044 \expandafter\MT@ex@do
1045 \fi
1046 }
```

`\MT@ex@split`

```
1047 \def\MT@ex@split#1=#2=#3\relax{%
1048 \def\@tempa{#1}%
1049 \ifx\@tempa\@empty \else
1050 \MT@get@slot
1051 \ifnum\MT@char > \m@ne
1052 \@tempcntb=#2\relax
```

Take an optional factor into account.

```
1053 \ifnum\MT@ex@factor@=\@m \else
1054 \MT@scale\@tempcntb \MT@ex@factor@ \@m
1055 \fi
1056 \ifnum\@tempcntb > \MT@ex@max
1057 \MT@warn@ex@too@large\MT@ex@max
1058 \else
1059 \ifnum\@tempcntb < \MT@ex@min
1060 \MT@warn@ex@too@large\MT@ex@min
1061 \fi
1062 \fi
1063 \efcode\MT@font\MT@char=\@tempcntb
1064 <debug>\MT@info@n1{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#2]}%
```

Heirs, heirs, I love thy heirs.

```
1065 \MT@ifdefined@c@T\MT@ex@inh@name{%
1066 \MT@ifdefined@n@T{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1067 \expandafter\MT@map@tlist@c
1068 \csname MT@inh@\MT@ex@inh@name @\MT@char @\endcsname
1069 \MT@set@ex@heirs
1070 }%
1071 }%
1072 \fi
```

```

1073 \fi
1074 }

\MT@warn@ex@too@large

1075 \def\MT@warn@ex@too@large#1{%
1076 \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1077 character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1078 Setting it to the maximum of \number#1}%
1079 \@tempcntb=#1\relax
1080 }

\MT@get@ex@opt Apply different values to this font?
\MT@ex@factor@ 1081 \def\MT@get@ex@opt{%
\MT@stretch@ 1082 \MT@set@listname
1083 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1084 \MT@let@cn\MT@ex@factor@{MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1085 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 1086 }{%
1087 \let\MT@ex@factor@\MT@ex@factor
1088 }%
1089 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1090 \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1091 \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
1092 \def\@tempa{autoexpand}%
1093 \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1094 \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @preset}{%
1095 \MT@preset@ex
1096 \let\MT@reset@ef@codes\relax
1097 }%
1098 }

\MT@get@ex@opt@

1099 \def\MT@get@ex@opt@#1#2{%
1100 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}{%
1101 \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
1102 \MT@vinfo{... : #2}%
1103 }{%
1104 \MT@let@nn{MT@#1@}{MT@#1}%
1105 }%
1106 }

\MT@set@ex@heirs

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

\MT@preset@ex

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

```

### 13.4.3 Interword Space (Glue)

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

```

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

```

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

```

1125 \def\MT@set@sp@codes{%
1126   \MT@if@list@exists{%
1127     \MT@get@dimen@six
1128     \MT@get@opt
1129     \MT@reset@sp@codes
1130     \MT@get@inh@list
1131     \MT@set@inputenc{c}%
1132     \MT@load@list\MT@sp@cc@name
1133     \MT@set@listname
1134     \MT@let@cn\@tempc{\MT@sp@cc@\MT@sp@cc@name}%
1135     \expandafter\MT@sp@do\@tempc,\relax,%
1136   }\MT@reset@sp@codes
1137 }

```

`\MT@sp@do`

```

1138 \def\MT@sp@do#1,{%
1139   \ifx\relax#1\@empty \else
1140     \MT@sp@split #1=\relax
1141     \expandafter\MT@sp@do
1142   \fi
1143 }

```

`\MT@sp@split`

```

1144 \def\MT@sp@split#1=#2=#3\relax{%
1145   \def\@tempa{#1}%
1146   \ifx\@tempa\@empty \else
1147     \MT@get@slot
1148     \ifnum\MT@char > \m@ne
1149       \MT@get@char@unit
1150       \MT@sp@split@val#2\relax
1151     \fi
1152   \fi
1153 }

```

`\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).

```

1154 \def\MT@sp@split@val#1,#2,#3\relax{%
1155   \def\@tempb{#1}%
1156   \MT@ifempty\@tempb\relax{%
1157     \MT@get@space@unit\tw@
1158     \MT@scale@to@em
1159     \knbscode\MT@font\MT@char=\@tempcntb
1160     <debug>\MT@info@n1{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1161   }%
1162   \def\@tempb{#2}%
1163   \MT@ifempty\@tempb\relax{%
1164     \MT@get@space@unit\thr@@
1165     \MT@scale@to@em
1166     \stbscode\MT@font\MT@char=\@tempcntb
1167     <debug>\MT@info@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1168   }%

```

```

1169 \def\@tempb{#3}%
1170 \MT@ifempty\@tempb\relax{%
1171   \MT@get@space@unit4%
1172   \MT@scale@to@em
1173   \shbscode\MT@font\MT@char=\@tempcntb
1174 <debug>\MT@info@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1175 }%
1176 \MT@ifdefined@c@T\MT@sp@inh@name{%
1177   \MT@ifdefined@n@T\MT@inh@\MT@sp@inh@name @\MT@char @}%
1178   \expandafter\MT@map@tlist@c
1179   \csname MT@inh@\MT@sp@inh@name @\MT@char @\endcsname
1180   \MT@set@sp@heirs
1181 }%
1182 }%
1183 }

\MT@set@sp@heirs

1184 \def\MT@set@sp@heirs#1{%
1185   \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1186   \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1187   \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1188 <*debug>
1189   \MT@info@n1{2}{-- heir of \MT@char: #1}%
1190   \MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1191                                     \number\stbscode\MT@font\MT@char/%
1192                                     \number\shbscode\MT@font\MT@char}%
1193 </debug>
1194 }

\MT@set@all@sp
\MT@reset@sp@codes 1195 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1196 <debug>\MT@info@n1{3}{-- knbs/stbs/shbs: setting all to \number#1/\number#2/\number#3}%
1197   \MT@do@font{%
1198     \knbscode\MT@font\@tempcnta=#1\relax
1199     \stbscode\MT@font\@tempcnta=#2\relax
1200     \shbscode\MT@font\@tempcnta=#3\relax
1201   }%
1202 }
1203 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1204 \let\MT@reset@sp@codes\relax

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

```

```

1225 \fi
1226 \MT@set@all@sp\@tempa\@tempc\@tempb
1227 }

```

### 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.

```

1228 \MT@requires@pdfTeX6{
1229   \def\MT@kerning{\MT@maybe@do{kn}}
1230 }{
1231   \let\MT@kerning\relax
1232 }

```

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

```

1233 \def\MT@set@kn@codes{%
1234   \MT@if@list@exists{%
1235     \MT@get@dimen@six
1236     \MT@get@opt
1237     \MT@reset@kn@codes
1238     \MT@get@inh@list
1239     \MT@set@inputenc{c}%
1240     \MT@load@list\MT@kn@c@name
1241     \MT@set@listname
1242     \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
1243     \expandafter\MT@kn@do\@tempc,\relax,%
1244   }\MT@reset@kn@codes
1245 }

```

`\MT@kn@do`

```

1246 \def\MT@kn@do#1,{%
1247   \ifx\relax#1\@empty \else
1248     \MT@kn@split #1=\relax
1249     \expandafter\MT@kn@do
1250   \fi
1251 }

```

`\MT@kn@split`

```

1252 \def\MT@kn@split#1=#2=#3\relax{%
1253   \def\@tempa{#1}%
1254   \ifx\@tempa\@empty \else
1255     \MT@get@slot
1256     \ifnum\MT@char > \m@ne
1257       \MT@get@char@unit
1258       \MT@kn@split@val#2\relax
1259     \fi
1260   \fi
1261 }

```

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

```

1262 \def\MT@kn@split@val#1,#2\relax{%
1263   \def\@tempb{#1}%
1264   \MT@ifempty\@tempb\relax{%
1265     \MT@get@space@unit\tw@
1266     \MT@scale@to@em
1267     \knbbcode\MT@font\MT@char=\@tempcntb
1268     (debug) \MT@info{n\{4\};;; knbc (\MT@char): \number\knbbcode\MT@font\MT@char: [#1]}%
1269   }%
1270   \def\@tempb{#2}%
1271   \MT@ifempty\@tempb\relax{%
1272     \MT@get@space@unit\tw@
1273     \MT@scale@to@em

```

```

1274 \knaccode\MT@font\MT@char=\@tempcntb
1275 <debug>\MT@info@n1{4}{;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1276 }%
1277 \MT@ifdefined@c@T\MT@kn@inh@name{%
1278 \MT@ifdefined@n@T{MT@inh@MT@kn@inh@name @\MT@char @}{%
1279 \expandafter\MT@map@tlist@c
1280 \csname MT@inh@MT@kn@inh@name @\MT@char @\endcsname
1281 \MT@set@kn@heirs
1282 }%
1283 }%
1284 }

\MT@set@kn@heirs

1285 \def\MT@set@kn@heirs#1{%
1286 \knbccode\MT@font#1=\knbccode\MT@font\MT@char
1287 \knaccode\MT@font#1=\knaccode\MT@font\MT@char
1288 <*debug>
1289 \MT@info@n1{2}{-- heir of \MT@char: #1}%
1290 \MT@info@n1{4}{;; knbc (#1): \number\knbccode\MT@font\MT@char/%
1291 \number\knaccode\MT@font\MT@char}%
1292 </debug>
1293 }

\MT@set@all@kn
\MT@reset@kn@codes 1294 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1295 <debug>\MT@info@n1{3}{-- knac/knbc: setting all to \number#1/\number#2}%
1296 \MT@do@font{%
1297 \knbccode\MT@font\@tempcnta=#1\relax
1298 \knaccode\MT@font\@tempcnta=#2\relax
1299 }%
1300 }
1301 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1302 \let\MT@reset@kn@codes\relax

\MT@preset@kn
\MT@preset@kn@ 1303 \def\MT@preset@kn{%
1304 \expandafter\expandafter\expandafter\MT@preset@kn@
1305 \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1306 }
1307 \def\MT@preset@kn@#1,#2\@nil{%
1308 \ifx\MT@kn@unit@\empty
1309 \MT@warn@preset@tewidth{kn}%
1310 \let\MT@preset@aux\MT@preset@aux@factor
1311 \else
1312 \let\MT@preset@aux\MT@preset@aux@space
1313 \fi

Letterspacing factor.
1314 \MT@ifstreq\MT@kn@context{letterspacing}{%
1315 \@tempcnta\MT@kn@factor@\relax
1316 \MT@scale\@tempcnta \MT@letterspacing@ \@m
1317 \edef\MT@kn@factor@{\number\@tempcnta}%
1318 }\relax
1319 \MT@preset@aux{#1}\@tempa
1320 \MT@preset@aux{#2}\@tempb
1321 \MT@set@all@kn\@tempa\@tempb
1322 }

```

### 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.

```
1323 \MT@requires@pdftex6{
1324   \DeclareRobustCommand\lsstyle{%
1325     \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`.

```
1326     \begingroup
1327       \escapechar\m@ne
1328       \expandafter\MT@exp@two@n\expandafter\MT@in@clist\expandafter
1329       {\expandafter\string\expandafter
1330        \csname\curr@fontshape/\f@size\expandafter\endcsname}%
1331       {\csname MT@knlist@font@\MT@kn@setname\endcsname}%
1332       \ifMT@inlist@ \else
1333         \expandafter\MT@xadd
1334         \csname MT@knlist@font@\MT@kn@setname\endcsname
1335         {\expandafter\string\csname\curr@fontshape/\f@size\endcsname,}%
1336       \fi
1337     \endgroup
1338     \fi
1339     \MT@kerningtrue
1340     \pdfappendkern\@ne
1341     \pdfprependkern\@ne
1342     \def\MT@kn@setname{all}%
1343     \MT@ifdefined@c@TF\MT@letterspacing@\relax{%
1344       \let\MT@letterspacing@\MT@letterspacing
1345     }%
1346     \microtypecontext{kerning=letterspacing}%
1347   }
1348 }{
1349   \DeclareRobustCommand\lsstyle{%
1350     \MT@warning{Letterspacing only works with pdftex version 1.40\MessageBreak
1351       or newer. You might want to use the 'soul' package\MessageBreak
1352       instead}%
1353     \MT@gl@et\lsstyle\relax
1354   }
1355 }
```

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

```
1356 \DeclareRobustCommand\textls[2][ ]{%
1357   \MT@ifempty{#1}%
1358   {\let\MT@letterspacing@\@undefined}%
1359   {\KV@sp@def\MT@letterspacing@{#1 } }%
1360   {\lsstyle #2}%
1361 }
1362 /beta
```

### 13.4.6 Disabling Ligatures

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

```
1363 \MT@requires@pdftex5{
1364   \def\MT@noligatures{%
1365     \ifMT@noligatures
1366       \MT@dotrue
1367       \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1368         \MT@ifdefined@n@TF{MT@checklist@#1}%
```

```

1369      {\csname MT@checklist@##1\endcsname}%
1370      {\MT@checklist@{##1}}%
1371      {n1}%
1372      }%
1373      \else
1374      \MT@dofalse
1375      \fi
1376      \ifMT@do
1377      \pdfnoligatures\MT@font
1378      \MT@vinfo{... Disabling ligatures}%
1379      \fi
1380      }
1381      }{
1382      \let\MT@noligatures\relax
1383      }

```

### 13.4.7 Loading the Configuration

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

```

1384 \def\MT@load@list#1{%
1385   \edef\@tempa{#1}%
1386   \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa @load}%
1387   \MT@ifstreq\@tempa\@tempb{%
1388     \MT@warning{\@nameuse{MT@abbr@\MT@feat} list `@\@tempa' cannot load itself}%
1389   }{%
1390     \ifx\@tempb\relax \else
1391       \MT@ifdefined@n@TF{MT@MT@feat @c@\@tempb}{%
1392         \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list `@\@tempb'}%
1393         \begin@group
1394         \MT@load@list\@tempb
1395         \end@group
1396         \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list
1397           \noexpand\MessageBreak`@\@tempb'}%
1398         \MT@let@cn\@tempc{MT@MT@feat @c@\@tempb}%
1399         \expandafter\csname MT@MT@feat @do\expandafter\endcsname\@tempc,\relax,%
1400       }{%
1401         \MT@warning{\@nameuse{MT@abbr@\MT@feat} list `@\@tempb' undefined.
1402           Cannot load\MessageBreak it from list `@\@tempa'}%
1403       }%
1404     \fi
1405   }%
1406 }

```

`\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.

```

1407 \let\MT@file@list\empty
1408 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

1409 \MT@in@clist{#1}\MT@file@list
1410 \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.

```

1411 \MT@begin@catcodes
1412 \let\MT@begin@catcodes\relax
1413 \let\MT@end@catcodes\relax
1414 \InputIfFileExists{mt-#1.cfg}{%
1415   \edef\MT@curr@file{mt-#1.cfg}%
1416   \MT@vinfo{... Loading configuration file \MT@curr@file}%

```



```

1417 \MT@xadd\MT@file@list{#1,}%
1418 }{%
1419 \expandafter\MT@get@basefamily#1\relax\relax\relax
1420 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1421 \ifMT@inlist@else
1422 \InputIfFileExists{mt-\@tempa.cfg}{%
1423 \edef\MT@curr@file{mt-\@tempa.cfg}%
1424 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1425 \MT@xadd\MT@file@list{\@tempa,#1,}%
1426 }{%
1427 \MT@vinfo{... No configuration file mt-#1.cfg}%
1428 \MT@xadd\MT@file@list{#1,}%
1429 }%
1430 \fi
1431 }%
1432 \endgroup
1433 \fi
1434 }

```

`\MT@begin@catcodes` We have to make sure that all characters have the correct category code. Especially, new lines and spaces should be ignored, since files might be loaded in the middle of the document. This is basically `\nfss@catcodes` (from the  $\text{\LaTeX}$  kernel). I've added: & (in tabulars), !, ?, , , : (french), ,, \$, -, ~, and = (Turkish babel).

OK, now all printable characters up to 127 are 'other'. We hope that letters are always letters and numbers other. (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.

```

1435 \def\MT@begin@catcodes{%
1436 \begingroup
1437 \makeatletter
1438 \catcode\^7%
1439 \catcode\ 9%
1440 \catcode\^^I9%
1441 \catcode\^^M9%
1442 \catcode\\\z0
1443 \catcode\{\@ne
1444 \catcode\}\tw@
1445 \catcode\#6%
1446 \catcode\%14%
1447 \MT@map@tlist@n
1448 {\!\\"\$&\'\(\)\*+,\-\.\/\:\;\<=\>\?[\]\_-\`|/~}%
1449 \@makeoother

```

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

```

1450 \let\KV@@sp@def\def
1451 }

```

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

```

1452 \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 ...).

```

1453 \def\MT@get@basefamily#1#2#3#4\relax{%
1454 \ifx#2\relax \def\@tempa{#1}\else
1455 \ifx#3\relax \def\@tempa{#1#2}\else
1456 \def\@tempa{#1#2#3}%

```

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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```

1457 \ifx\relax#4\relax \else
1458 \MT@ifstreq{#4}{\string x}\relax{%
1459 \MT@ifstreq{#4}{\string j}\relax{%
1460 \MT@ifstreq{#4}{\string w}\relax{%
1461 \def\@tempa{#1#2#3#4}}}\fi\fi\fi
1462 }

```

`\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@ 1463 \def\MT@get@listname#1{%
1464 <debug>\MT@info{n}{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
1465 \let\MT@listname\undefined
1466 \def\@tempb{#1}%
1467 \MT@map@tlist@c\MT@try@order\MT@get@listname@
1468 }
1469 \def\MT@get@listname@#1{%
1470 \expandafter\MT@next@listname#1%
1471 \ifx\MT@listname\undefined \else
1472 \expandafter\MT@tlist@break
1473 \fi
1474 }

```

`\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.

```

1475 \def\MT@try@order{%
1476 {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
1477 {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
1478 }

```

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

```

1479 \def\MT@next@listname#1#2#3#4{%
1480 \edef\@tempa{\MT@encoding
1481 /\ifnum#1=\@ne \MT@family\fi
1482 /\ifnum#2=\@ne \MT@series\fi
1483 /\ifnum#3=\@ne \MT@shape\fi
1484 /\ifnum#4=\@ne *\fi
1485 \MT@context}%
1486 <debug>\MT@info{n}{1}{trying \@tempa}%
1487 \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
1488 \MT@next@listname@#4%
1489 }{%

```

Also try with an alias family.

```

1490 \ifnum#1=\@ne
1491 \ifx\MT@familyalias\@empty \else
1492 \edef\@tempa{\MT@encoding
1493 /\MT@familyalias
1494 /\ifnum#2=\@ne \MT@series\fi
1495 /\ifnum#3=\@ne \MT@shape\fi

```

```

1496             /\ifnum#4=\@ne *\fi
1497             \MT@context}%
1498 <debug>\MT@info@n1{1}{(alias) \@tempa}%
1499             \MT@ifdefined@nT{MT@\@tempb @\@tempa}{%
1500             \MT@next@listname@#4%
1501             }%
1502         \fi
1503     \fi
1504 }%
1505 }

```

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

```

1506 \def\MT@next@listname@#1{%
1507     \ifnum#1=\@ne
1508         \expandafter\MT@in@rlist\csname MT@\@tempb @\@tempa @sizes\endcsname
1509         \ifMT@inlist@
1510             \let\MT@listname\MT@size@name
1511         \fi
1512     \else
1513         \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
1514     \fi
1515 }

```

\MT@if@list@exists

```

\MT@context 1516 \def\MT@if@list@exists{%
1517     \MT@let@cn\MT@context{MT@\MT@feat @context}%
1518     \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
1519     \MT@get@listname{\MT@feat @c}%
1520     \MT@ifdefined@c@TF\MT@listname{%
1521         \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
1522         \ifMT@nonselected
1523             \MT@vinfo{... Applying non-selected expansion (list `\'MT@listname')}%
1524         \else
1525             \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `\'MT@listname'}%
1526         \fi
1527         \@firstoftwo
1528     }{%

```

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

```

1529     \MT@let@nc{MT@\MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

1530     \ifMT@nonselected
1531         \MT@vinfo{... Applying non-selected expansion}%
1532     \else
1533         \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
1534             for font\MessageBreak`\'MT@@font'%
1535             \ifx\MT@context\@empty\else\space(context: `\'MT@context')}\fi.
1536         Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
1537     \fi
1538     \@secondoftwo
1539 }%
1540 }

```

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

```

\MT@context 1541 \def\MT@get@inh@list{%
1542     \let\MT@context\@empty
1543     \MT@get@listname{\MT@feat @inh}%
1544     \MT@ifdefined@c@TF\MT@listname{%
1545         \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
1546 <*debug>

```

```

1547 \MT@info{n1}{1}{... Using \nameuse{MT@abbr@MT@feat} inheritance list
1548 ~\MT@listname'}%
1549 </debug>
1550 \MT@let@cn\@tempc{MT@MT@feat @inh@MT@listname}%

If the list is \@empty, it has already been parsed.
1551 \ifx\@tempc\@empty \else
1552 <debug>\MT@info{n1}{1}{parsing inheritance list ...}%

The group is only required in case an input encoding is given.
1553 \begingroup
1554 \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak~\MT@listname'}%
1555 \MT@set@inputenc{inh}%
1556 \expandafter\MT@inh@do\@tempc,\relax,%
1557 \global\MT@let@nc{MT@MT@feat @inh@MT@listname}\@empty
1558 \endgroup
1559 \fi
1560 }{%
1561 \MT@let@nc{MT@MT@feat @inh@name}\@undefined
1562 }%
1563 }

```

### 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.

```

1564 \def\MT@get@slot{%
1565 \escapechar~\
1566 \let\MT@char@m@ne
1567 \MT@noresettrue

```

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

```

1568 \MT@toks=\expandafter{\@tempa}%

```

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

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

```

1569 \expandafter\MT@is@letter\@tempa\relax\relax
1570 \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.

```

1571 \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.

```

1572      \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
1573      \MT@is@symbol

```

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

```

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

```

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

```

1576      \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1577      \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
1578      \fi
1579      \fi

1580      \let\MT@char\MT@char@
1581      \ifnum\MT@char < \z@
1582      \MT@warn@unknown
1583      \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.

```

1584      \ifMT@norest \else
1585      \MT@warn@rest
1586      \let\MT@char@m@ne
1587      \fi
1588      \fi
1589      \escapechar@m@ne
1590 }

```

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

```

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

```

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

```

1593 \def\MT@is@letter#1#2\relax{%
1594   \ifcat a\noexpand#1\relax
1595   \edef\MT@char@{\number`#1}%
1596   \ifx\#2\%
1597   (debug) \MT@info@n{3}{> '\the\MT@toks' is a letter (\MT@char@)}%
1598   \else
1599   \MT@norestfalse
1600   \fi
1601   \else
1602   \ifcat !\noexpand#1\relax
1603   \edef\MT@char@{\number`#1}%
1604   (debug) \MT@info@n{3}{> '\the\MT@toks' is a character (\MT@char@)}%
1605   \ifx\#2\%
1606   \ifnum\MT@char@>127 \MT@warn@ascii \fi
1607   \else
1608   \MT@norestfalse
1609   \expandafter\MT@is@number#1#2\relax\relax
1610   \fi
1611   \fi
1612   \fi
1613 }

```

\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.

```

1614 \def\MT@is@number#1#2#3\relax{%
1615   \ifx\relax#3\relax \else
1616     \ifx\relax#2\relax \else
1617       \MT@noresttrue
1618       \if#1"\relax
1619         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
1620 (debug)\MT@info@n1{3}{> ... a hexadecimal number: \MT@char@}%
1621       \else
1622         \if#1'\relax
1623           \def\MT@char@{\number#1#2#3}%
1624 (debug)\MT@info@n1{3}{> ... an octal number: \MT@char@}%
1625         \else
1626           \MT@ifint{#1#2#3}{%
1627             \def\MT@char@{\number#1#2#3}%
1628 (debug)\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
1629           }\MT@norestfalse
1630       \fi
1631     \fi
1632     \ifnum\MT@char@ > \cc1v
1633       \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
1634       \let\MT@char@\m@ne
1635     \fi
1636   \fi
1637 \fi
1638 }

```

`\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.

```
1639 \def\MT@is@active#1#2\@nil{%
1640   \ifnum\catcode`#1 = \active
1641     \begingroup
1642       \setdisplay@protect
1643       \let\IeC\@firstofone
1644       \let\@inpc@undefined\MT@undefined@char
```

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

```
1645   \def\UTFviii@defined##1{\ifx ##1\relax
1646     \MT@undefined@char{utf8}\else\expandafter##1\fi}%
1647   \edef\x{%
1648     \def\noexpand\@tempa{\@tempa}%
```

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

```
1649     \MT@toks={\the\MT@toks\space(= \@tempa)}%
1650   }%
1651   \expandafter\endgroup\x
1652 \fi
1653 }
```

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

```
1654 \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.

```
1655 \def\MT@is@symbol{%
1656   \edef\MT@char{\csname\MT@encoding\MT@detokenizec\@tempa\endcsname}%
1657   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1658     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
1659   \ifnum\MT@char@ < \z@
```

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

```
1660   \expandafter\MT@is@letter\MT@char\relax\relax
1661 \fi
1662 }
```

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

```

\MT@charstring 1663 \begingroup
1664   \catcode`\/=0
1665   /MT@map@tlist@n{\/C/H/A/R}/@makeother
1666   /lowercase{%
1667     /def/x{%
1668       /def/MT@charstring{\CHAR"%
1669       /def/MT@is@char##1\CHAR"##2##3##4/relax{%
1670         /ifx/relax##1/relax
1671         /if##3\relax
1672         /edef/MT@char@{/number"##2}%
1673         /MT@testrest/MT@charstring{##3##4}%
1674         /else
1675         /edef/MT@char@{/number"##2##3}%
1676         /MT@testrest/MT@charstring{##4}%
1677         /fi
1678         (debug) /MT@info@n1{3}{> `the/MT@toks' is a \char (/MT@char@)}%
1679         /fi
1680       }%
1681     }%
1682   }
1683 /expandafter/endgroup/x

```

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

```

1684 \def\MT@is@composite#1#2\relax{%
1685   \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`.

```

1686   \edef\MT@char{\expandafter
1687     \csname\expandafter
1688       \string\csname\MT@encoding\endcsname
1689       \MT@detokenize@n{#1}-%
1690       \MT@detokenize@n{#2}%
1691       \endcsname}%
1692   \expandafter\MT@is@letter\MT@char\relax\relax
1693   \fi
1694 }

```

(What about math? Well, for a moment the following looked like a solution, with `\mt@is@mathchar` defined accordingly, analogous to `\MT@is@char` above, to pick up the last two tokens (the `\meaning` of a `\mathchardef`'ed command expands to its hexadecimal notation):

```

\def\MT@is@mathchar#1{%
  \if\relax\noexpand#1% it's a macro
    \let\x#1%
  \else % it's a character
    \mathchardef\x=\mathcode~#1\relax
  \fi
  \expandafter\MT@exp@two@c\expandafter\mt@is@mathchar\expandafter
  \meaning\expandafter\x\mt@mathcharstring\relax\relax\relax
}

```

However, the problem is that `\mathcodes` and `\mathchardefs` have global scope. Therefore, if they are changed by a package that loads different math fonts, there is no guarantee whatsoever that things will still be correct (e. g., the minus in `cmsy`



when the euler package is loaded). So, no way to go, unfortunately.)

Some warning messages, for performance reasons separated here.

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

```
\MT@set@listname 1695 \def\MT@set@listname{%
1696   \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list\noexpand\MessageBreak
1697     \@nameuse{MT@MT@feat @c@name}}}%
1698 }
```

`\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.

```
1699 \def\MT@warn@ascii{%
1700   \MT@warning@n1{Character \the\MT@toks' (= \MT@char@)
1701     is outside of ASCII range.\MessageBreak
1702     You must load the 'inputenc' package before using\MessageBreak
1703     8-bit characters in \MT@curr@list@name}%
1704 }
```

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

```
1705 \def\MT@warn@number@too@large#1{%
1706   \MT@warning@n1{%
1707     Number #1 in encoding \MT@encoding' too large!\MessageBreak
1708     Ignoring it in \MT@curr@list@name}%
1709 }
```

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

```
1710 \def\MT@warn@rest{%
1711   \MT@warning@n1{%
1712     Unknown slot number of character\MessageBreak\the\MT@toks'%
1713     \MT@warn@maybe@inputenc\MessageBreak
1714     in font encoding \MT@encoding'.\MessageBreak
1715     Make sure it's a single character\MessageBreak
1716     (or a number) in \MT@curr@list@name}%
1717 }
```

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

```
1718 \def\MT@warn@unknown{%
1719   \MT@warning@n1{%
1720     Unknown slot number of character\MessageBreak\the\MT@toks'%
1721     \MT@warn@maybe@inputenc\MessageBreak
1722     in font encoding \MT@encoding' in \MT@curr@list@name}%
1723 }
```

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

```
1724 \def\MT@warn@maybe@inputenc{%
1725   \MT@ifdefined@n@T
1726   {MT@MT@feat @MT@cat @\csname MT@MT@feat @MT@cat @name\endcsname @inputenc}%
1727   { (input encoding \@nameuse
1728     {MT@MT@feat @MT@cat @\csname MT@MT@feat @MT@cat @name\endcsname @inputenc})}%
1729 }
```

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

Up to version 1.3 of this package, we were using `\define@newfont` as the hook, which is only called for *new* fonts, and therefore seemed the natural choice. However, this meant that we had to take special care to catch all fonts: we additionally had to set up the default font, the error font (if it wasn't the default font), we had to check for some packages that might have been loaded before microtype and were loading fonts, e. g., jurabib, ledmac, pi font (loaded by hyperref), tipa, and probably many more. Furthermore, we had to include a hack for the IEEEtran class which loads all fonts in the class file itself (to fine tune inter-word spacing). Then I learned that even my favourite class, memoir, loads fonts. To cut this short: it seemed to get out of hand, and I decided that it would be better to use `\pickup@font` and decide for ourselves whether we've already seen that font. I hope the overhead isn't too large.

`\MT@font@list` We use a comma separated list.

```
1730 \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.

```
1731 \let\MT@font\empty
1732 \MT@addto@setup{%
1733   \g@addto@macro\do@subst@correction{%
1734     \xdef\MT@font{\csname \curr@fontshape/\f@size\endcsname}%
1735   }
```

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

```
1736 \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
1737 \ifx\pickup@font\MT@orig@pickupfont \else
1738   \MT@warning@nl{%
1739     Command \string\pickup@font\space is not defined as expected.\MessageBreak
1740     Double-check whether micro-typography is indeed.\MessageBreak
1741     applied to the document.\MessageBreak (Hint: Turn on `verbose' mode)%
1742   }
1743 \fi
```

Then we append our stuff. Everything is done inside a group.

```
1744 \g@addto@macro\pickup@font{\begingroup}
```

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

```
1745 \MT@with@package{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
1746 \g@addto@macro\pickup@font{%
1747     \escapechar\m@ne
```

If `\MT@font` is empty, no substitution has taken place, hence `\font@name` is correct. Otherwise, if they are different, `\font@name` does not describe the font actually used. This test will catch first order substitutions, like `\bx` to `\b`, but it will still fail if the substituting font is itself substituted.

```
1748     \ifx\MT@font\@empty
1749     \let\MT@font\font@name
1750     \else
1751     \ifx\MT@font\font@name \else
1752     \MT@register@subst@font
1753     \fi
1754 \fi
```

Then, check whether we've already seen the font.

```
1755 \MT@check@font
```

If not, set it up and register it.

```
1756 \ifMT@inlist@ \else
1757 \MT@setupfont
1758 \MT@register@font
1759 \fi
1760 \MT@glet\MT@font\@empty
1761 \endgroup
1762 }
```

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

```
1763 \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.

```
1764 \let\MT@orig@add@accent\add@accent
1765 \def\add@accent#1#2{%
1766     \let\pickup@font\MT@orig@pickupfont
1767     \MT@orig@add@accent{#1}{#2}%
1768     \let\pickup@font\MT@pickupfont
1769 }
1770 }
```

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

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

```
1771 \def\MT@check@font{\MT@exp@one@n\MT@in@cl@ist\MT@font\MT@font@list}
```

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

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

```

1772 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
\MT@register@font Register the current font.
1773 \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.

```
1774 \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.

```

1775 \def\MT@check@font@cx{%
1776   \MT@if@true
1777   \MT@map@clist@c\MT@active@features{%
1778     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
1779     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1780     \ifMT@inlist@
1781       \MT@let@c{MT@\@nameuse{MT@abbr@##1}}\relax
1782     \else
1783       \MT@if@false
1784     \fi
1785   }%
1786   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
1787 }

```

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

```

1788 \def\MT@register@subst@font@cx{%
1789   \MT@map@clist@c\MT@active@features{%
1790     \expandafter\MT@xadd
1791     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1792     {\font@name,}%
1793   }%
1794 }

```

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

```

1795 \def\MT@register@font@cx{%
1796   \MT@map@clist@c\MT@active@features{%
1797     \expandafter\ifx\csname MT@\@nameuse{MT@abbr@##1}\endcsname\relax\else
1798     \expandafter\MT@xadd
1799     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1800     {\MT@font,}%
1801     \def\@tempa{##1}%
1802     \expandafter\MT@map@tlist@c
1803     \csname MT@##1@doc@contexts\endcsname
1804     \MT@rem@from@lists
1805   \fi
1806   }%
1807 }

```

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

```

1808 \def\MT@rem@from@lists#1{%
1809   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
1810     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist
1811     \expandafter\MT@font\csname MT@\@tempa @#1font@list\endcsname

```

```
1812 }%
1813 }
```

`\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.

```
1814 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
1815 \MT@addto@setup{%
1816   \def\microtypecontext#1{%
1817     \MT@setup@contexts
1818     \let\MT@reset@context\relax
1819     \setkeys{MTC}{#1}%
1820     \selectfont
1821     \MT@reset@context
1822   }%
1823 }
```

`\MT@reset@context` We have to reset the font at the end of the group, provided there actually was a change.

```
\MT@reset@context@
1824 \def\MT@reset@context@{%
1825   \MT@vinfo{<<< Resetting contexts\on@line
1826   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
1827   <debug&beta> / \MT@kn@context/\MT@sp@context
1828   }%
1829   \selectfont
1830 }
```

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

```
1831 \def\MT@setup@contexts{%
1832   \MT@map@clist@c\MT@active@features
1833   {\global\MT@let@nc{MT@##1@font@list}\MT@font@list}%
1834   \MT@glet\MT@check@font\MT@check@font@cx
1835   \MT@glet\MT@register@font\MT@register@font@cx
1836   \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
1837   \MT@glet\MT@setup@contexts\relax
1838 }
```

`\MT@define@context`

```
1839 \def\MT@define@context#1{%
1840   \define@key{MTC}{#1}[] {%
1841     \KV@sp@def\@tempb{#1}%
1842     \edef\@tempb{\@nameuse{MT@rbba@\@tempb}}%
1843     \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
1844     \ifMT@inlist@
```

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

```
1845   \MT@ifempty{##1}{\def\MT@val{}}{\KV@sp@def\MT@val{##1}}%
1846   \expandafter\ifx\csname MT@\@tempb @context\endcsname\MT@val
1847   <debug> \MT@dinfo{1}{>>> no change of #1 context: ` \MT@val' }%
1848   \else
1849     \MT@vinfo{>>> Changing #1 context to ` \MT@val' \MessageBreak\on@line
1850   <debug> \space{previous: ` \@nameuse{MT@\@tempb @context}' }%
1851   }%
1852   \def\MT@reset@context{\aftergroup\MT@reset@context@}%
```

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

```
1853   \global\MT@let@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes@}%
```

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

```

1854 \expandafter\MT@exp@one@n\expandafter\MT@in@tlist
1855 \expandafter\MT@val\csname MT@\@tempb @doc@contexts\endcsname
1856 \ifMT@inlist@ \else
1857 \expandafter\MT@xadd\csname MT@\@tempb @doc@contexts\endcsname{{\MT@val}}%
1858 (debug) \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
1859 \fi
1860 \MT@edef@n{MT@\@tempb @context}{\MT@val}%
1861 \fi
1862 \fi
1863 }%
1864 }
1865 \MT@map@clist@c\MT@features@long{\MT@define@context{#1}}

```

\MT@pr@context Initialize the contexts.

\MT@ex@context 1866 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%}

\MT@sp@context 1867 \MT@def@n{MT@#1@context}{@}%

\MT@kn@context 1868 \MT@def@n{MT@#1@doc@contexts}{@}%

1869 }

\MT@pr@doc@contexts 1870 \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 all available features will be created.

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

```

1871 \def\DeclareMicrotypeSet{%
1872 \ifstar
1873 {\ifnextchar[\MT@DeclareSetAndUseIt
1874 {\MT@DeclareSetAndUseIt[]}}%
1875 {\ifnextchar[\MT@DeclareSet
1876 {\MT@DeclareSet[]}}%
1877 }

```

\MT@DeclareSet

\MT@DeclareSetAndUseIt 1878 \def\MT@DeclareSet[#1]{%

1879 \MT@DeclareSet@{#1}%

1880 }

1881 \def\MT@DeclareSetAndUseIt[#1]#2#3{%

1882 \MT@DeclareSet@{#1}{#2}{#3}%

1883 \UseMicrotypeSet[#1]{#2}%

1884 }

\MT@DeclareSet@

1885 \def\MT@DeclareSet@#1#2#3{%

1886 \KV@sp@def\@tempa{#1}%

1887 \MT@ifempty\@tempa{%

1888 \MT@map@clist@c\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%

1889 }{%

1890 \MT@map@clist@c\@tempa{%

1891 \KV@sp@def\@tempa{##1}%

1892 \MT@ifempty\@tempa\relax{%

1893 \MT@is@feature{set declaration `#2'}{%

1894 \MT@exp@one@n\MT@declare@sets

```

1895         {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
1896     }%
1897 }%
1898 }}%
1899 }%
1900 }

```

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

```
1901 \let\MT@curr@set@name\@empty
```

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

```

1902 \def\MT@declare@sets#1#2#3{%
1903   \KV@sp@def\MT@curr@set@name{#2}%
1904   \MT@ifdefined@nT{MT@#1@set@{\MT@curr@set@name}{%
1905     \MT@warning{Redefining set `~\MT@curr@set@name'}%
1906   }%
1907   \global\MT@let@nc{MT@#1@set@{\MT@curr@set@name}\@empty
1908   <debug>\MT@info{1}{declaring ~\nameuse{MT@abbr@#1} set ~\MT@curr@set@name'}%
1909   \setkeys{MT@#1@set}{#3}%
1910 }

```

\MT@define@set@keys Define the keyval keys for font sets.

```

1911 \def\MT@define@set@keys#1{%
1912   \MT@define@set@key@{encoding}{#1}%
1913   \MT@define@set@key@{family}{#1}%
1914   \MT@define@set@key@{series}{#1}%
1915   \MT@define@set@key@{shape}{#1}%
1916   \MT@define@set@key@size{#1}%
1917   \MT@define@set@key@font{#1}%
1918 }

```

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

```

1919 \def\MT@define@set@key@#1#2{%
1920   \define@key{MT@#2@set}{#1}[]{}%
1921   \global\MT@let@nc{MT@#2list@#1@{\MT@curr@set@name}\@empty
1922   \MT@map@clist@n{##1}{%
1923     \KV@sp@def\MT@val{####1}%
1924     \MT@get@highlevel{#1}%

```

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

```

1925     \MT@exp@two@n@g@addto@macro
1926     {\csname MT@#2list@#1@{\MT@curr@set@name\expandafter\endcsname}%
1927     {\MT@val,}%
1928   }%

```

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

```

1929   \edef\x{%
1930     \expandafter\noexpand\expandafter\MT@fix@font@spec
1931     \expandafter\noexpand\csname MT@#2list@#1@{\MT@curr@set@name\endcsname
1932   }%
1933   \MT@exp@one@n\MT@addto@setup\x
1934   <debug>\MT@info@n1{1}{-- #1: ~\nameuse{MT@#2list@#1@{\MT@curr@set@name}}%
1935   }%
1936 }

```

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

```

1937 \def\MT@get@highlevel#1{%
1938   \expandafter\MT@test@ast\MT@val*\@nil%

```

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

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

```
1940 \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.

```
1941 }%
```

```
1942 }
```

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

```
1943 \def\MT@test@ast#1*#2\@nil{%
```

```
1944 \def\@tempa{#1}%
```

```
1945 \MT@ifempty{#2}\@gobble\@firstofone
```

```
1946 }
```

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

```
1947 \def\MT@fix@font@spec#1{%
```

```
1948 \xdef#1{#1}%
```

```
1949 \global\MT@make@string#1%
```

```
1950 }
```

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

```
1951 \def\MT@define@set@key@size#1{%
```

```
1952 \define@key{MT@#1@set}{size}[]{%
```

```
1953 \MT@map@clist@n{##1}{%
```

```
1954 \KV@sp@def\MT@val{###1}%
```

```
1955 \expandafter\MT@get@range\MT@val--\@nil
```

```
1956 \ifx\MT@val\relax \else
```

```
1957 \expandafter\MT@xadd
```

```
1958 \csname MT@#1list@size@\MT@curr@set@name\endcsname
```

```
1959 {{{\MT@lower}{\MT@upper}\relax}}%
```

```
1960 \fi
```

```
1961 }%
```

```
1962 <debug>\MT@info@n1{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
```

```
1963 }%
```

```
1964 }
```

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  $\{\langle lower\ bound \rangle\} \{\langle upper\ bound \rangle\} \{\langle list\ name \rangle\}$ .

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

**\MT@lower** 1965 \def\MT@get@range#1-#2-#3\@nil{%

```
1966 \MT@ifempty{#1}{%
```

```
1967 \MT@ifempty{#2}{%
```

```
1968 \let\MT@val\relax
```

```
1969 }{%
```

```
1970 \def\MT@lower{0}%
```

```
1971 \def\MT@val{#2}%
```

```
1972 \MT@get@size
```

```
1973 \edef\MT@upper{\MT@val}%
```

```
1974 }%
```

```
1975 }{%
```

```
1976 \def\MT@val{#1}%
```

```
1977 \MT@get@size
```

```
1978 \ifx\MT@val\relax \else
```

```
1979 \edef\MT@lower{\MT@val}%
```

```
1980 \MT@ifempty{#2}{%
```

```
1981 \MT@ifempty{#3}%
```

```
1982 {\def\MT@upper{-1}}%
```



2048pt is TeX's maximum font size.

```

1983     {\def\MT@upper{2048}}%
1984   }{%
1985     \def\MT@val{#2}%
1986     \MT@get@size
1987     \ifx\MT@val\relax \else
1988       \MT@ifdim\MT@lower>\MT@val{%
1989         \MT@warning{%
1990           Invalid size range (\MT@lower\space > \MT@val) in font set
1991           ~\MT@curr@set@name'.\MessageBreak Swapping sizes}%
1992         \edef\MT@upper{\MT@lower}%
1993         \edef\MT@lower{\MT@val}%
1994       }{%
1995         \edef\MT@upper{\MT@val}%
1996       }%
1997       \MT@ifdim\MT@lower=\MT@upper
1998       {\def\MT@upper{-1}}%
1999       \relax
2000     \fi
2001   }%
2002 \fi
2003 }%
2004 }
```

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

```
2005 \def\MT@get@size{%
```

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

```

2006   \if*\MT@val\relax
2007     \def\@tempa{\normalsize}%
2008   \else
2009     \MT@let@cn\@tempa{\MT@val}%
2010   \fi
2011   \ifx\@tempa\relax \else
```

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

```

2012   \begingroup
2013     \def\set@fontsize##1##2##3##4\@nil{\gdef\MT@val{##2}}%
2014     \@tempa\@nil
2015   \endgroup
2016 \fi
```

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

```

2017   \MT@ifdimen\MT@val{%
2018     \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2019     \edef\MT@val{\strip@pt\@tempdima}%
2020   }{%
2021     \MT@warning{Could not parse font size ~\MT@val'\MessageBreak
2022       in font set ~\MT@curr@set@name'}%
2023     \let\MT@val\relax
2024   }%
2025 }
```

\MT@define@set@key@font

```
2026 \def\MT@define@set@key@font#1{%
```

```

2027 \define@key{MT@#1@set}{font}[]{%
2028   \global\MT@let@nc{MT@#1list@font@MT@curr@set@name}\@empty
2029   \MT@map@clist@n{##1}{%
2030     \KV@sp@def\MT@val{###1}%
2031     \expandafter\MT@get@font\MT@val////\@nil
2032     \MT@exp@two@n@g@addto@macro
2033     {\csname MT@#1list@font@MT@curr@set@name\expandafter\endcsname}%
2034     {\MT@val,}%
2035   }%
2036   \edef\x{%
2037     \expandafter\noexpand\expandafter\MT@fix@font@spec
2038     \expandafter\noexpand\csname MT@#1list@font@MT@curr@set@name\endcsname
2039   }%
2040   \MT@exp@one@n\MT@addto@setup\x
2041   <debug>\MT@info@n{1}{-- font: \@nameuse{MT@#1list@font@MT@curr@set@name}}%
2042   }%
2043 }

```

\MT@get@font Translate any asterisks.

```

2044 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2045   \MT@ifempty{#1#2#3#4#5}\relax{%
2046     \let\@tempb\@empty
2047     \def\MT@temp{#1/#2/#3/#4/#5}%
2048     \MT@get@axis{encoding}{#1}%
2049     \MT@get@axis{family}{#2}%
2050     \MT@get@axis{series}{#3}%
2051     \MT@get@axis{shape}{#4}%
2052     \MT@ifempty{#5}{%
2053       \MT@warn@axis@empty{size}{\string\normal size}%
2054       \def\MT@val{*}%
2055     }{%
2056       \def\MT@val{#5}%
2057     }%
2058     \MT@get@size
2059     \ifx\MT@val\relax\def\MT@val{0}\fi
2060     \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2061     \let\MT@val\@tempb
2062   }%
2063 }

```

\MT@get@axis

```

2064 \def\MT@get@axis#1#2{%
2065   \def\MT@val{#2}%
2066   \MT@get@highlevel{#1}%
2067   \MT@ifempty\MT@val{%
2068     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2069     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2070   }\relax
2071   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2072 }

```

\MT@warn@axis@empty

```

2073 \def\MT@warn@axis@empty#1#2{%
2074   \MT@warning{#1 axis is empty in font specification\MessageBreak
2075     ~\MT@temp'. Using ~#2' instead}%
2076 }

```

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

```

2077 \MT@map@clist@c\MT@features{\MT@define@set@keys{#1}}

```

It is also used for \DisableLigatures.

```

2078 \MT@define@set@keys{n}

```

`\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.

```

2079 \renewcommand*\UseMicrotypeSet[2][]{%
2080   \KV@sp@def\@tempa{#1}%
2081   \MT@ifempty\@tempa{%
2082     \MT@map@clistc\MT@features{{\MT@use@set{##1}{#2}}}%
2083   }%
2084   \MT@map@clistc\@tempa{%
2085     \KV@sp@def\@tempa{#1}%
2086     \MT@ifempty\@tempa\relax{%
2087       \MT@is@feature{activation of set `#2'}{%
2088         \MT@exp@one@n\MT@use@set
2089           {csname MT@rbb@\@tempa\endcsname}{#2}%
2090       }%
2091     }%
2092   }%
2093 }%
2094 }

```

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

```

\MT@ex@setname 2095 \def\MT@use@set#1#2{%
\MT@use@set 2096   \KV@sp@def\@tempa{#2}%
2097   \MT@ifdefined@n@TF{MT@#1@set@{\@tempa}}{%
2098     \MT@xdef@n{MT@#1@setname}{\@tempa}%
2099     \MT@info{Using \@nameuse{MT@abbr@#1} set \@tempa'}%
2100   }%
2101   \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2102     \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2103   }%
2104   \MT@warning{%
2105     The \@nameuse{MT@abbr@#1} set \@tempa' is undeclared.\MessageBreak
2106     Using set \@nameuse{MT@#1@setname}' instead}%
2107   }%
2108 }

```

`\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'.

```

2109 \MT@requires@pdfTeX5{
2110   \renewcommand*\DisableLigatures[1]{%
2111     \edef\MT@active@features{\MT@active@features,nl}%
2112     \MT@noligaturestrue
2113     \MT@declare@sets{nl}{no ligatures}{#1}%
2114     \gdef\MT@nl@setname{no ligatures}%
2115   }
2116 }{

```

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

```

2117   \renewcommand*\DisableLigatures[1]{%
2118     \MT@warning{Disabling ligatures of a font is only possible\MessageBreak
2119       with pdfTeX version 1.30 or newer.\MessageBreak
2120       Ignoring \string\DisableLigatures}%
2121     \MT@glet\DisableLigatures\@gobble
2122   }
2123 }

```

`\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.

```

2124 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2125   \KV@sp@def\@tempa{#1}%
2126   \MT@ifempty\@tempa{%
2127     \MT@map@clistc\MT@features{{\MT@set@default@set{##1}{#2}}}%

```

```

2128 }{%
2129 \MT@map@c\list@c\@tempa{%
2130 \KV@esp@def\@tempa{#1}%
2131 \MT@ifempty\@tempa\relax{%
2132 \MT@is@feature{declaration of default set `#2'}{%
2133 \MT@exp@one@n\MT@set@default@set
2134 {\csname MT@rbba@\@tempa\endcsname}{#2}%
2135 }%
2136 }%
2137 }}%
2138 }%
2139 }

\MT@default@pr@set
\MT@default@ex@set 2140 \def\MT@set@default@set#1#2{%
2141 \KV@esp@def\@tempa{#2}%
\MT@default@kn@set 2142 \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@default@sp@set 2143 (debug)\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set \@tempa'}%
\MT@set@default@set 2144 \MT@xdef@n{MT@default@#1@set}{\@tempa}%
2145 }{%
2146 \MT@warning{%
2147 The \@nameuse{MT@abbr@#1} set \@tempa' is not declared.\MessageBreak
2148 Cannot make it the default set. Using set\MessageBreak `all' instead}%
2149 \MT@xdef@n{MT@default@#1@set}{all}%
2150 }%
2151 }

```

`\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.

```

2152 \renewcommand*\DeclareMicrotypeAlias[2]{%
2153 \KV@esp@def\@tempa{#1}%
2154 \KV@esp@def\@tempb{#2}%
2155 \MT@make@string\@tempb
2156 \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2157 \MT@warning{Alias font family \@tempb' will override
2158 alias \@nameuse{MT@\@tempa @alias}'\MessageBreak
2159 for font family \@tempa'}}%
2160 \MT@xdef@n{MT@\@tempa @alias}{\@tempb}%

```

If we encounter this command while a font is being set up, we also set the alias for the current font so that if `\DeclareMicrotypeAlias` has been issued inside a configuration file, the configuration file for the alias font will be loaded, too.

```

2161 \MT@ifdefined@c@T\MT@family{%
2162 (debug)\MT@dinfo{1}{Activating alias font \@tempb' for \@tempa'}%
2163 \MT@gl@t\MT@familyalias\@tempb
2164 }%
2165 }

```

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

```

2166 \def\LoadMicrotypeFile#1{%
2167 \KV@esp@def\@tempa{#1}%
2168 \MT@make@string\@tempa
2169 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2170 \ifMT@inlist@
2171 \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2172 \else
2173 \MT@xadd\MT@file@list{\@tempa,}%
2174 \MT@begin@catcodes
2175 \InputIfFileExists{mt-\@tempa.cfg}{%
2176 \edef\MT@curr@file{mt-\@tempa.cfg}%
2177 \MT@vinfo{... Loading configuration file \MT@curr@file}%

```

```

2178 }{%
2179 \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2180           does not exist}%
2181 }%
2182 \MT@end@catcodes
2183 \fi
2184 }

```

### 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.

```

2185 <#beta>
2186 \def\DeclareMicrotypeBabelHook#1#2{%
2187 \MT@map@clist@n{#1}{%
2188 \KV@sp@def\@tempa{##1}%
2189 \MT@gdef@n{MT@babel@\@tempa}{#2}%
2190 }%
2191 }
2192 </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.

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

```

2193 \def\SetProtrusion{%
2194 \MT@begin@catcodes
2195 \MT@SetProtrusion
2196 }

```

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

```

\MT@pr@c@name 2197 \newcommand*\MT@SetProtrusion[3][]{%

```

```

\MT@extra@context 2198 \let\MT@extra@context\@empty

```

`\MT@permute@list` Parse the optional first argument. We first have to know the name before we can deal with the extra options.

```

2199 \MT@set@named@keys{MT@pr@c@}{#1}%
2200 <debug> \MT@dinfo{1}{creating protrusion list `~\MT@pr@c@name'}%
2201 \def\MT@permute@list{pr@c}%
2202 \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>`, ...

```

2203 \MT@permute

```

... which we can now define to be `<#3>`. Here, as elsewhere, we have to make the definitions global, since they will occur inside a group.

```

2204 \MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}%
2205 \MT@end@catcodes
2206 }

```

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

```
2207 \def\SetExpansion{%
2208   \MT@begin@catcodes
2209   \MT@SetExpansion
2210 }
```

\MT@SetExpansion

```
\MT@ex@c@name 2211 \newcommand*\MT@SetExpansion[3][{}]{%
\MT@extra@context 2212   \let\MT@extra@context\@empty
\MT@permutelist 2213   \MT@set@named@keys{MT@ex@c}{#1}%
2214   \MT@ifdefined@n@T{MT@ex@c@MT@ex@c@name @factor}{%
2215     \ifnum\csname MT@ex@c@MT@ex@c@name @factor\endcsname>\@m
2216       \MT@warning@n{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
2217         too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
2218         maximum of 1000}%
2219       \global\MT@let@nc{MT@ex@c@MT@ex@c@name @factor}\@m
2220     \fi
2221   }%
2222   <debug>\MT@dinfo{1}{creating expansion list `~\MT@ex@c@name'}%
2223   \def\MT@permutelist{ex@c}%
2224   \setkeys{MT@cfg}{#2}%
2225   \MT@permute
2226   \MT@gdef@n{MT@ex@c@MT@ex@c@name}{#3}%
2227   \MT@end@catcodes
2228 }
2229 <beta>
```

\SetExtraSpacing

```
2230 \def\SetExtraSpacing{%
2231   \MT@begin@catcodes
2232   \MT@SetExtraSpacing
2233 }
```

\MT@SetExtraSpacing

```
\MT@sp@c@name 2234 \newcommand*\MT@SetExtraSpacing[3][{}]{%
\MT@extra@context 2235   \let\MT@extra@context\@empty
\MT@permutelist 2236   \MT@set@named@keys{MT@sp@c}{#1}%
2237   <debug>\MT@dinfo{1}{creating spacing list `~\MT@sp@c@name'}%
2238   \def\MT@permutelist{sp@c}%
2239   \setkeys{MT@cfg}{#2}%
2240   \MT@permute
2241   \MT@gdef@n{MT@sp@c@MT@sp@c@name}{#3}%
2242   \MT@end@catcodes
2243 }
```

\SetExtraKerning

```
2244 \def\SetExtraKerning{%
2245   \MT@begin@catcodes
2246   \MT@SetExtraKerning
2247 }
```

\MT@SetExtraKerning

```
\MT@kn@c@name 2248 \newcommand*\MT@SetExtraKerning[3][{}]{%
\MT@extra@context 2249   \let\MT@extra@context\@empty
\MT@permutelist 2250   \MT@set@named@keys{MT@kn@c}{#1}%
2251   <debug>\MT@dinfo{1}{creating kerning list `~\MT@kn@c@name'}%
2252   \def\MT@permutelist{kn@c}%
2253   \setkeys{MT@cfg}{#2}%
2254   \MT@permute
2255   \MT@gdef@n{MT@kn@c@MT@kn@c@name}{#3}%

```

```
2256 \MT@end@catcodes
2257 }
```

```
2258 </beta>
```

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

\MT@options

```
2259 \def\MT@set@named@keys#1#2{%
2260   \def\x##1name=##2,##3\@nil{%
2261     \setkeys{#1}{name=##2}%
2262     \def\MT@options{##1##3}%
2263     \MT@rem@from@clist{name=}\MT@options
2264   }%
2265   \x#2,name=,\@nil
2266   \@expandtwoargs\setkeys{#1}\MT@options
2267 }
```

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

```
2268 \def\MT@define@code@key#1#2{%
2269   \define@key{MT@#2}{#1}[]{%
2270     \@tempcnta=\@ne
2271     \MT@map@clist@n{##1}{%
2272       \KV@sp@def\MT@val{####1}%

```

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

```
2273     \MT@get@highlevel{#1}%
2274     \MT@edef@n{MT@temp#1the\@tempcnta}{\MT@val}%
2275     \advance\@tempcnta \@ne
2276   }%
2277 }%
2278 }
```

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

```
2279 \def\MT@define@code@key@size#1{%
2280   \define@key{MT@#1}{size}[]{%
2281     \MT@map@clist@n{##1}{%
2282       \KV@sp@def\MT@val{####1}%
2283       \expandafter\MT@get@range\MT@val--\@nil
2284       \ifx\MT@val\relax \else
2285         \expandafter\MT@xadd\csname MT@tempsize\endcsname
2286         {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2287       \fi
2288     }%
2289   }%
2290 }
```

\MT@define@code@key@font

```
2291 \def\MT@define@code@key@font#1{%
2292   \define@key{MT@#1}{font}[]{%
2293     \MT@map@clist@n{##1}{%
2294       \KV@sp@def\MT@val{####1}%
2295       \expandafter\MT@get@font@and@size\MT@val////\@nil
2296       \MT@xdef@n{MT@MT@permutelist @\@tempb\MT@extra@context}%
2297       {\csname MT@MT@permutelist @name\endcsname}%
2298     < *debug >
2299     \MT@info@n1{1}{initializing: use list for font \@tempb=\MT@val
2300       \ifx\MT@extra@context\@empty\else\MessageBreak
2301         (context: \MT@extra@context)\fi}%
2302     </debug >
2303     \expandafter\MT@xaddb
2304     \csname MT@MT@permutelist @\@tempb\MT@extra@context @sizes\endcsname
```

```

2305      {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
2306    }%
2307  }%
2308 }

```

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

```

2309 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2310   \MT@ifempty{#1#2#3#4#5}\relax{%
2311     \let\@tempb\@empty
2312     \def\MT@temp{#1/#2/#3/#4/#5}%
2313     \MT@get@axis{encoding}{#1}%
2314     \MT@get@axis{family}{#2}%
2315     \MT@get@axis{series}{#3}%
2316     \MT@get@axis{shape}{#4}%

```

Append an asterisk for the size.

```

2317   \edef\@tempb{\@tempb*}%
2318   \MT@ifempty{#5}{%
2319     \MT@warn@axis@empty{size}{\string\normalsize}%
2320     \def\MT@val{*}%
2321   }{%
2322     \def\MT@val{#5}%
2323   }%
2324   \MT@get@size
2325 }%
2326 }

2327 \MT@define@code@key{encoding}{cfg}
2328 \MT@define@code@key{family}{cfg}
2329 \MT@define@code@key{series}{cfg}
2330 \MT@define@code@key{shape}{cfg}
2331 \MT@define@code@key@size{cfg}
2332 \MT@define@code@key@font{cfg}

```

`\MT@define@opt@keys` The options in the optional first argument.

```

2333 \def\MT@define@opt@keys#1{%

```

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

```

2334   \define@key{MT@#1@c}{name}[]{}%
2335   \MT@ifempty{#1}{%
2336     \MT@edefn{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2337   }{%
2338     \MT@edefn{MT@#1@c@name}{#1}%
2339     \MT@ifdefinedn@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
2340       \MT@warning{Redefining list `\'@nameuse{MT@#1@c@name}'}%
2341     }%
2342   }%
2343   \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
2344 }%
2345 \MT@define@opt@key{#1}{load}%
2346 \MT@define@opt@key{#1}{factor}%
2347 \MT@define@opt@key{#1}{preset}%
2348 \MT@define@opt@key{#1}{inputenc}%

```

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

```

2349   \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
2350 }

```

`\MT@define@opt@key`

```

2351 \def\MT@define@opt@key#1#2{%
2352   \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%

```



```

2353 \MT@xdefn{MT@#1@c@MT@curr@set@name @#2}{##1}}%
2354 }

```

```

2355 \MT@map@clist@c\MT@features{\MT@define@opt@keys{#1}}

```

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

```

2356 \define@key{MT@pr@c}{unit}[character]{%
2357 \global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
2358 \def\@tempa{#1}%
2359 \MT@ifstreq\@tempa{relative}{%
2360 \MT@warning{Value `relative' for key `unit' is deprecated.\MessageBreak
2361 Use `unit=character' instead. For now, I'll do it.\MessageBreak
2362 for you}%
2363 \def\@tempa{character}%
2364 }\relax
2365 \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.

```

2366 \MT@ifdimen\@tempa{%
2367 \global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa
2368 }{%
2369 \MT@warning{\@tempa' is not a dimension.\MessageBreak
2370 Ignoring it and setting values relative to.\MessageBreak
2371 character widths}%
2372 }%
2373 }%
2374 }

```

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

```

2375 <beta>
2376 \def\MT@define@key@unit#1{%
2377 \define@key{MT@#1@c}{unit}[space]{%
2378 \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
2379 \def\@tempa{##1}%
2380 \MT@ifstreq\@tempa{character}\relax{%
2381 \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
2382 \MT@ifstreq\@tempa{space}\relax{%
2383 \MT@ifdimen\@tempa{%
2384 \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa
2385 }{%
2386 \MT@warning{\@tempa' is not a dimension.\MessageBreak
2387 Ignoring it and setting values relative to.\MessageBreak
2388 width of space}%
2389 }%
2390 }%
2391 }%
2392 }%
2393 }
2394 \MT@define@key@unit{sp}
2395 \MT@define@key@unit{kn}
2396 </beta>

```

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

```

2397 \def\MT@define@ex@opt@key#1{%
2398 \define@key{MT@ex@c}{#1}[]{%
2399 \MT@ifempty{##1}\relax{%
2400 \MT@ifint{##1}{%

```

A space terminates the number.

```

2401 \MT@gdefn{MT@ex@c@MT@curr@set@name @#1}{##1 }%
2402 }%

```

```

2403 \MT@warning{%
2404     Value `##1' for option `#1' is not a number.\MessageBreak
2405     Ignoring it}%
2406 }%
2407 }%
2408 }%
2409 }

2410 \MT@define@ex@opt@key{stretch}
2411 \MT@define@ex@opt@key{shrink}
2412 \MT@define@ex@opt@key{step}
2413 \define@key{MT@ex@c}{auto}[true]{%
2414     \def\@tempa{#1}%
2415     \csname if\@tempa\endcsname

```

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

```

2416 \MT@requires@pdftex4{%
2417     \MT@gdefn{MT@ex@c@MT@curr@set@name @auto}{autoexpand}%
2418 }{%
2419     \MT@warning{pdftex too old for automatic font expansion}%
2420 }
2421 \else
2422     \MT@requires@pdftex4{%
2423         \global\MT@letenc{MT@ex@c@MT@curr@set@name @auto}\@empty
2424     }\relax
2425 \fi
2426 }

```

### 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.

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

```

2427 \renewcommand*\DeclareCharacterInheritance[1][]{%
2428     \let\MT@extra@context\@empty
2429     \let\MT@extra@inputenc\@undefined
2430     \let\MT@inh@feat\@empty
2431     \setkeys{MT@inh@}{#1}%
2432     \MT@begin@catcodes
2433     \MT@set@inh@list
2434 }

```

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

```

2435 \def\MT@set@inh@list#1#2{%
2436     \MT@ifempty\MT@inh@feat{%
2437         \MT@map@clistc\MT@features{\MT@declare@char@inh{##1}{#1}{#2}}}%
2438     }{%
2439         \MT@map@clistc\MT@inh@feat{%
2440             \KV@sp@def\@tempa{#1}%
2441             \MT@ifempty\@tempa\relax{%
2442                 \MT@exp@one@n\MT@declare@char@inh
2443                 {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
2444             }%
2445         }}%

```

```

2446 }%
2447 \MT@end@catcodes
2448 }

```

The keys for the optional argument.

```

2449 \MT@map@clist@c\MT@features@long{%
2450 \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}
2451 \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}

```

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

```

2452 \def\MT@declare@char@inh#1#2#3{%
2453 \MT@edefn{MT@#1@inh@name}%
2454 {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
2455 \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
2456 \MT@ifdefined@c@T\MT@extra@inputenc{%
2457 \MT@xdefn{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
2458 <bug>\MT@info{1}{creating inheritance list ~\@nameuse{MT@#1@inh@name}}%
2459 \MT@gdefn{MT@#1@inh@\cename MT@#1@inh@name\endcsname}{#3}%
2460 \def\MT@permute@list{#1@inh}%
2461 \setkeys{MT@inh}{#2}%
2462 \MT@permute
2463 }

```

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

```

2464 \define@key{MT@inh}{encoding}[]{%
2465 \def\MT@val{#1}%
2466 \expandafter\MT@encoding@check\MT@val,\@nil
2467 \MT@get@highlevel{encoding}%
2468 \MT@edefn{MT@tempencoding1}{\MT@val}%
2469 }

```

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

```

2470 \def\MT@encoding@check#1,#2\@nil{%
2471 \MT@ifempty{#2}\relax{%
2472 \edef\MT@val{#1}%
2473 \MT@warning{You may only specify one encoding for character\MessageBreak
2474 inheritance lists. Ignoring encoding(s) #2}%
2475 }%
2476 }

```

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

```

2477 \MT@define@code@key{family}{inh}
2478 \MT@define@code@key{series}{inh}
2479 \MT@define@code@key{shape}{inh}
2480 \MT@define@code@key{size}{inh}
2481 \MT@define@code@key{font}{inh}

```

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

```

2482 \def\MT@inh@do#1,{%
2483 \ifx\relax#1\@empty \else
2484 \MT@inh@split #1==\relax
2485 \expandafter\MT@inh@do
2486 \fi
2487 }

```

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

```

2488 \def\MT@inh@split#1=#2=#3\relax{%
2489   \def\@tempa{#1}%
2490   \ifx\@tempa\@empty \else
2491     \MT@get@slot
2492     \ifnum\MT@char > \m@ne
2493       \let\MT@val\MT@char
2494       \MT@map@clist@n{#2}{%
2495         \def\@tempa{##1}%
2496         \ifx\@tempa\@empty \else
2497           \MT@get@slot
2498           \ifnum\MT@char > \m@ne
2499             \expandafter\MT@xadd
2500             \csname MT@inh@\MT@listname @\MT@val @\endcsname
2501             {{\MT@char}}%
2502           \fi
2503         \fi
2504       }%
2505   <debug>
2506   \MT@info@n{2}{children of #1 (\MT@val):
2507     \nameuse{MT@inh@\MT@listname @\MT@val @}}%
2508 </debug>
2509   \fi
2510 \fi
2511 }

```

### 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.

```

2512 \def\MT@permute{%
2513   \let\MT@cnt@encoding\@ne
2514   \MT@permute@

```

Undefine commands for the next round.

```

2515   \MT@permute@reset
2516 }
2517 \def\MT@permute@{%
2518   \let\MT@cnt@family\@ne
2519   \MT@permute@@
2520   \MT@increment\MT@cnt@encoding
2521   \MT@ifdefined@nT{MT@tempencoding\MT@cnt@encoding}%
2522   \MT@permute@
2523 }
2524 \def\MT@permute@@{%
2525   \let\MT@cnt@series\@ne
2526   \MT@permute@@@
2527   \MT@increment\MT@cnt@family
2528   \MT@ifdefined@nT{MT@tempfamily\MT@cnt@family}%
2529   \MT@permute@@
2530 }
2531 \def\MT@permute@@@{%
2532   \let\MT@cnt@shape\@ne
2533   \MT@permute@@@@
2534   \MT@increment\MT@cnt@series

```

```

2535 \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
2536 \MT@permute@@@
2537 }
2538 \def\MT@permute@@@{%
2539 \MT@permute@@@@
2540 \MT@increment\MT@cnt@shape
2541 \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
2542 \MT@permute@@@
2543 }

```

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

```

2544 \def\MT@permute@@@@{%
2545 \MT@permute@define{encoding}%
2546 \ifMT@document
2547 \ifx\MT@tempencoding\empty \else
2548 \MT@ifdefined@n@TF{T@MT@tempencoding}\relax
2549 {\expandafter\expandafter\expandafter@gobble}%
2550 \fi
2551 \fi
2552 \MT@permute@@@@
2553 }

```

\MT@permute@@@@

```

2554 \def\MT@permute@@@@{%
2555 \MT@permute@define{family}%
2556 \MT@permute@define{series}%
2557 \MT@permute@define{shape}%
2558 \edef\@tempa{\MT@tempencoding
2559 \MT@tempfamily
2560 \MT@tempseries
2561 \MT@tempshape
2562 \MT@ifdefined@c@T\MT@tempsize *}%

```

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

```

2563 \def\@tempb{////}%
2564 \ifx\@tempa\@tempb \else
2565 \ifx\MT@tempencoding\empty
2566 \MT@warning{%
2567 You have to specify an encoding for\MessageBreak
2568 \@nameuse{MT@abbr@MT@permutelist} list
2569 ~\@nameuse{MT@MT@permutelist @name}'.\MessageBreak
2570 Ignoring it}%
2571 \else
2572 \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

2573 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
2574 \MT@map@tlist@c
2575 \MT@tempsize
2576 \MT@check@rlist
2577 }%
2578 \expandafter\MT@xaddb
2579 \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
2580 \MT@tempsize
2581 <*debug>
2582 \MT@info@n1{1}{initializing: use list for font \@tempa,\MessageBreak
2583 sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
2584 @sizes\endcsname}%
2585 </debug>
2586 }%

```

Only one list should apply to a given combination.

```

2587 \MT@ifdefined@n@T{MT@MT@permute@list @\tempa\MT@extra@context}{%
2588 \MT@warning{\@nameuse{MT@abbr@MT@permute@list} list
2589 \@nameuse{MT@MT@permute@list @name}' will override list\MessageBreak
2590 \@nameuse{MT@MT@permute@list @\tempa\MT@extra@context}'
2591 for font \@tempa'}%
2592 }%
2593 {*debug}
2594 \MT@info@n@l{1}{initializing: use list for font \tempa
2595 \ifx\MT@extra@context\empty\else\MessageBreak
2596 (context: \MT@extra@context)\fi}%
2597 {/debug}
2598 }%
2599 \MT@xdef@n{MT@MT@permute@list @\tempa\MT@extra@context}%
2600 {\csname MT@MT@permute@list @name\endcsname}%
2601 \fi
2602 \fi
2603 }

```

\MT@permute@define Define the commands.

```

2604 \def\MT@permute@define#1{%
2605 \expandafter\@tempcnta=\csname MT@cnt@#1\endcsname\relax
2606 \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
2607 {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
2608 {\MT@let@nc{MT@temp#1}\empty}%
2609 }

```

\MT@permute@reset Reset the commands.

```

2610 \def\MT@permute@reset{%
2611 \MT@permute@reset@{encoding}%
2612 \MT@permute@reset@{family}%
2613 \MT@permute@reset@{series}%
2614 \MT@permute@reset@{shape}%
2615 \MT@glet\MT@tempsize\undefined
2616 }

```

\MT@permute@reset@

```

2617 \def\MT@permute@reset@#1{%
2618 \@tempcnta=\@ne
2619 \MT@loop
2620 \MT@let@nc{MT@temp#1\the\@tempcnta}\undefined
2621 \advance\@tempcnta\@ne
2622 \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
2623 \iftrue
2624 \iffalse
2625 \MT@repeat
2626 }

```

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

```

2627 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

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

```

2628 \def\MT@check@rlist@#1#2#3{%
2629 \def\@tempb{#1}%
2630 \def\@tempc{#2}%
2631 \MT@if@false
2632 \expandafter\MT@map@tlist@
2633 \csname MT@MT@permute@list @\tempa\MT@extra@context @size\endcsname
2634 \MT@check@range
2635 }

```

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

```
2636 \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.

```
2637 \def\MT@check@range@#1#2#3{%
```

```
2638   \MT@ifdim{#2}=\m@ne{%
```

```
2639   \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
2640       \MT@ifdim\@tempb={#1}\MT@iftrue\relax
```

```
2641   }{%
```

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

```
2642       \MT@ifdim\@tempb>{#1}\relax{%
```

```
2643       \MT@ifdim\@tempc>{#1}{%
```

```
2644       \MT@iftrue
```

```
2645       \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
```

```
2646       }\relax
```

```
2647   }%
```

```
2648   }%
```

```
2649   }{%
```

```
2650   \MT@ifdim\@tempc=\m@ne{%
```

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

```
2651       \MT@ifdim\@tempb<{#2}{%
```

```
2652       \MT@ifdim\@tempb<{#1}\relax\MT@iftrue
```

```
2653       }\relax
```

```
2654   }{%
```

- Both items are ranges.

```
2655       \MT@ifdim\@tempb<{#2}{%
```

```
2656       \MT@ifdim\@tempc>{#1}{%
```

```
2657       \MT@iftrue
```

```
2658       \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
```

```
2659       }\relax
```

```
2660       }\relax
```

```
2661   }%
```

```
2662   }%
```

```
2663   \ifMT@if@
```

```
2664       \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
```

```
2665       ~\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
```

```
2666       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.

```
2667       \expandafter\MT@tlist@break
```

```
2668   \fi
```

```
2669 }
```

## 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 2670 \newif\ifMT@opt@expansion
```

```
\ifMT@opt@DVI 2671 \newif\ifMT@opt@auto
```

```
2672 \newif\ifMT@opt@DVI
```

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

```

2673 \def\MT@define@option#1{%
2674   \define@key{MT}{#1}[true]{%
2675     \csname MT@opt@#1true\endcsname
2676     \MT@map@clist@n{##1}{%
2677       \KV@sp@def\MT@val{####1}%
2678       \MT@ifempty\MT@val\relax{%
2679         \csname MT@#1true\endcsname
2680         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2681         \MT@ifstreq\MT@val{true}\relax
2682         {%
2683           \MT@ifstreq\MT@val{false}{%
2684             \csname MT@#1false\endcsname
2685           }{%
2686             \MT@ifstreq\MT@val{compatibility}{%
2687               \MT@let@nc{MT@\@tempb @level}\@ne
2688             }{%
2689               \MT@ifstreq\MT@val{nocompatibility}{%
2690                 \MT@let@nc{MT@\@tempb @level}\tw@
2691               }{%

```

If everything failed, it should be a set name.

```

2692       \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
2693         \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
2694       }{%
2695         \MT@xdef@n{MT@\@tempb @setname}%
2696         {\@nameuse{MT@default@\@tempb @set}}%
2697         \MT@warning@n1{%
2698           The #1 set '\MT@val' is undeclared.\MessageBreak
2699           Using set '\@nameuse{MT@\@tempb @setname}' instead}%
2700       }%
2701     }%
2702   }%
2703 }%
2704 }%
2705 }%
2706 }%
2707 }%
2708 }

2709 \MT@define@option{protrusion}
2710 \MT@define@option{expansion}

```

`activate` is a shortcut for protrusion and expansion.

```

2711 \define@key{MT}{activate}[true]{%
2712   \setkeys{MT}{protrusion={#1}}%
2713   \setkeys{MT}{expansion={#1}}%
2714 }
2715 <beta>

```

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

```

2716 \def\MT@define@option@#1{%
2717   \define@key{MT}{#1}[true]{%
2718     \csname MT@opt@#1true\endcsname
2719     \MT@map@clist@n{##1}{%
2720       \KV@sp@def\MT@val{####1}%
2721       \MT@ifempty\MT@val\relax{%
2722         \csname MT@#1true\endcsname
2723         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2724         \MT@ifstreq\MT@val{true}\relax
2725       }%

```



```

2726 \MT@ifstreq\MT@val{false}{%
2727 \csname MT@#1false\endcsname
2728 }{%
2729 \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
2730 \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
2731 }{%
2732 \MT@xdef@n{MT@\@tempb @setname}%
2733 {\@nameuse{MT@default@\@tempb @set}}}%
2734 \MT@warning@n1{%
2735 The #1 set `\'MT@val' is undeclared.\MessageBreak
2736 Using set `\'@nameuse{MT@\@tempb @setname}' instead}%
2737 }%
2738 }%
2739 }%
2740 }%
2741 }%
2742 }%
2743 }
2744 \MT@define@option@{spacing}
2745 \MT@define@option@{kerning}
2746 /beta

```

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

```

2747 \def\MT@def@bool@opt#1#2{%
2748 \define@key{MT}{#1}[]{%
2749 \MT@ifempty{#1}{%
2750 {\def\@tempa{true}}%
2751 {\def\@tempa{#1}}%
2752 \MT@ifstreq\@tempa{true}\relax{%
2753 \MT@ifstreq\@tempa{false}\relax{%
2754 \MT@warning@n1{%
2755 `##1' is not an admissible value for option\MessageBreak
2756 `#1'. Assuming `false'}%
2757 \def\@tempa{false}%
2758 }%
2759 }%
2760 #2%
2761 }%
2762 }

```

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

```

2763 \def\MT@def@simple@bool@opt#1{\MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
2764 \MT@map@tlist@n{{draft}{auto}{selected}%
2765 /beta {babel}}%
2766 }\MT@def@simple@bool@opt

```

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

```

2767 \MT@def@bool@opt{DVIoutput}{%
2768 \csname if\@tempa\endcsname
2769 \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
2770 \pdfoutput\z@
2771 \else
2772 \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
2773 \pdfoutput\@ne
2774 \fi
2775 }

```

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.

```

2776 \MT@def@bool@opt{defersetup}{%
2777   \csname if\@tempa\endcsname \else
2778     \AtEndOfPackage{%
2779       \MT@setup@
2780       \let\MT@setup@\@empty
2781       \let\MT@addto@setup\@firstofone
2782     }%
2783 \fi
2784 }
```

`final` is the opposite to `draft`.

```

2785 \MT@def@bool@opt{final}{%
2786   \csname if\@tempa\endcsname
2787   \MT@draftfalse
2788 \else
2789   \MT@drafttrue
2790 \fi
2791 }
```

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

```

2792 \define@key{MT}{verbose}[]{%
2793   \let\MT@vinfo\MT@warn@err
2794   \MT@ifempty{#1}%
2795   {\def\@tempa{true}}%
2796   {\def\@tempa{#1}}%
2797   \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```

2798   \MT@ifstreq\@tempa{errors}{%
2799     \let\MT@warning\MT@warn@err
2800     \let\MT@warning@nl\MT@warn@err
2801   }{%
2802     \let\MT@vinfo\@gobble
2803     \MT@ifstreq\@tempa{false}\relax{%
2804       \MT@warning@nl{%
2805         `#1' is not an admissible value for option\MessageBreak
2806         `verbose'. Assuming `false'}%
2807     }%
2808   }%
2809 }%
2810 }
```

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

```

2811 \def\MT@def@num@opt#1{%
2812   \define@key{MT}{#1}[]{%
2813     \MT@ifempty{#1}%
2814     {\MT@let@cn\@tempa{MT@#1@default}}%
2815     {\def\@tempa{#1}}%
2816 }
```

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

```

2816   \MT@ifint\@tempa{%
2817     \MT@edef@n{MT@#1}{\@tempa}%
2818   }{\MT@warning@nl{%
2819     Value `##1' for option `#1' is not a number.\MessageBreak
2820     Using default value of \number\@nameuse{MT@#1@default}}%
2821   }%
2822 }
```

```

2823 }
2824 \MT@map@tlist@n{{stretch}{shrink}{step}%
2825 beta }{letterspacing}%
2826 }\MT@def@num@opt

factor will define the protrusion factor only.
2827 \define@key{MT}{factor}[]{}%
2828 \MT@ifempty{#1}%
2829 {\let\@tempa\MT@factor@default}%
2830 {\def\@tempa{#1}}%
2831 \MT@ifint\@tempa{%
2832 \edef\MT@pr@factor{\@tempa}%
2833 }{\MT@warning@nl{%
2834 Value `#1' for option `factor' is not a number.\MessageBreak
2835 Using default value of \number\MT@factor@default}%
2836 }%
2837 }

```

Unit for codes.

```

2838 \define@key{MT}{unit}[]{}%
2839 \MT@ifempty{#1}%
2840 {\def\@tempa{character}}%
2841 {\KV@@sp@def\@tempa{#1}}%
2842 \MT@ifstreq\@tempa{relative}{}%
2843 \MT@warning{Value `relative' for option `unit' is deprecated.\MessageBreak
2844 Use `unit=character' instead. For now, I'll do it.\MessageBreak
2845 for you}%
2846 \def\@tempa{character}%
2847 }\relax
2848 \MT@ifstreq\@tempa{character}\relax{%
2849 \MT@ifdimen\@tempa{%
2850 \let\MT@pr@unit\@tempa
2851 }{}%
2852 \MT@warning@nl{\@tempa' is not a dimension. Ignoring it and.\MessageBreak
2853 setting values relative to character widths}%
2854 }%
2855 }%
2856 }

```

### 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.)

```

2857 \MT@protrusiontrue
2858 \ifnum\pdfoutput<\@ne \else

```

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

```

2859 \MT@requires@pdftex4{
2860 \MT@expansiontrue
2861 \MT@autottrue
2862 }\relax
2863 \fi

```

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

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

```
2864 \define@key{MT}{config}[]{\relax}
2865 \def\MT@get@config#1config=#2,#3\@nil{%
2866   \MT@ifempty{#2}%
2867   {\def\MT@config@file{\MT@MT.cfg}}%
2868   {\KV@esp@def\MT@config@file{#2.cfg}}%
2869 }
2870 \expandafter\expandafter\expandafter\MT@get@config
2871 \csname opt@\currname.\@currentx\endcsname,config=\@nil
```

Load the file.

```
2872 \IfFileExists{\MT@config@file}{%
2873   \MT@info@nl{Loading configuration file \MT@config@file}%
2874   \MT@begin@catcodes
2875   \let\MT@begin@catcodes\relax
2876   \let\MT@end@catcodes\relax
2877   \let\MT@curr@file\MT@config@file
2878   \input{\MT@config@file}%
2879   \endgroup
2880 }{\MT@warning@nl{%
2881   Could not find configuration file '\MT@config@file'!\MessageBreak
2882   This will almost certainly cause undesired results.\MessageBreak
2883   Please fix your installation}%
2884 }
```

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

```
2885 \MT@map@clist@c\MT@features{%
2886   \MT@ifdefined@nTF{MT@default@#1@set}\relax
2887   {\MT@gdef@n{MT@default@#1@set}{all}}%
2888 }
```

### 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}
    {\gaddto@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.

```
2889 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
2890   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
2891   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
2892 \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.

```
2893 \def\microtypesetup{\setkeys{MT}}
2894 \MT@addto@setup{\def\microtypesetup{\setkeys{MTX}}}
2895 \def\MT@define@optionX#1#2{%
2896   \define@key{MTX}{#1}[true]{%
2897     \KV@sp@def\@tempb{#1}%
2898     \MT@map@clist@n{##1}{%
2899       \KV@sp@def\MT@val{###1}%
2900       \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
2901       \MT@ifempty\MT@val\relax{%
2902         \@tempcnta=\m@ne
2903         \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.

```
2904     \MT@checksetup\@tempb{%
2905       \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
2906       \MT@info{Enabling #1
2907         (level \number\csname MT@\@tempb @level\endcsname)}%
2908     }%
2909   }{%
2910     \MT@ifstreq\MT@val{false}{%
2911       \@tempcnta=\z@
2912       \MT@info{Disabling #1}%
2913     }{%
2914       \MT@ifstreq\MT@val{compatibility}{%
2915         \MT@checksetup\@tempb{%
2916           \@tempcnta=\@ne
2917           \MT@let@nc{MT@\@tempb @level}\@ne
2918           \MT@info{Setting #1 to level 1}%
2919         }%
2920       }{%
2921         \MT@ifstreq\MT@val{nocompatibility}{%
2922           \MT@checksetup\@tempb{%
2923             \@tempcnta=\tw@
2924             \MT@let@nc{MT@\@tempb @level}\tw@
2925             \MT@info{Setting #1 to level 2}%
2926           }%
2927         }{%
2928           \MT@warning{%
2929             Value '\MT@val' for key '#1' not recognized.\MessageBreak
2930             Use any of 'true', 'false', 'compatibility' or\MessageBreak
2931             'nocompatibility'%
2932           }%
2933         }%

```

```

2934     }%
2935   }%
2936   \ifnum\@tempcnta>\m@ne
2937     #2\@tempcnta\relax
2938   \fi
2939 }%
2940 }%
2941 }%
2942 }

```

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

```

2943 \def\MT@checksetup#1{%
2944   \expandafter\csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
2945   \expandafter\@firstofone
2946   \else
2947     \MT@warning{%
2948       You cannot enable \@nameuse{MT@abbr@#1} if it was disabled\MessageBreak
2949       in the package options,}%
2950   \expandafter\@gobble
2951   \fi
2952 }

2953 \MT@define@optionX{protrusion}\pdfprotrudechars
2954 \MT@define@optionX{expansion}\pdfadjustspacing
2955 <beta>

```

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

```

2956 \def\MT@define@optionX@#1#2{%
2957   \define@key{MTX}{#1}[true]{%
2958     \KV@esp@def\@tempb{#1}%
2959     \MT@map@clist@n{##1}{%
2960       \KV@esp@def\MT@val{####1}%
2961       \edef\@tempb{\csname MT@rbb@@\@tempb\endcsname}%
2962       \MT@ifempty\MT@val\relax{%
2963         \@tempcnta=\m@ne
2964         \MT@ifstreq\MT@val{true}{%
2965           \MT@checksetup\@tempb{%
2966             \@tempcnta=\@ne
2967             \MT@info{Enabling #1}%
2968           }%
2969         }{%
2970           \MT@ifstreq\MT@val{false}{%
2971             \@tempcnta=\z@
2972             \MT@info{Disabling #1}%
2973           }{%
2974             \MT@warning{%
2975               Value `~\MT@val' for key `~#1' not recognized.\MessageBreak
2976               Use either `true' or `false'}%
2977           }%
2978         }%
2979         \ifnum\@tempcnta>\m@ne
2980           #2\relax
2981         \fi
2982       }%
2983     }%
2984   }%
2985 }

2986 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
2987 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
2988   \pdfappendkern \@tempcnta}
2989 </beta>
2990 \define@key{MTX}{activate}[true]{%

```

```

2991 \setkeys{MTX}{protrusion={#1}}%
2992 \setkeys{MTX}{expansion={#1}}%
2993 }

```

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

```

2994 \let\MT@saved@setupfont\MT@setupfont
2995 \define@key{MTX}{disable}[]{}%
2996 \MT@info{Inactivate `\'MT@MT' package}%
2997 \let\MT@setupfont\@gobble
2998 }
2999 \define@key{MTX}{enable}[]{}%
3000 \MT@info{Reactivate `\'MT@MT' package}%
3001 \let\MT@setupfont\MT@saved@setupfont
3002 }

```

### 13.6.5 Processing the Options

`\MT@ProcessOptionsWithKV` Parse options.

```

3003 \def\MT@ProcessOptionsWithKV#1{%
3004   \let\@tempc\relax
3005   \let\KVo@tempa\@empty
3006   \MT@map@clist@c\@classoptionslist{%
3007     \def\CurrentOption{##1}%
3008     \MT@ifdefined@n@T{KV@#1@CurrentOption}{%
3009       \edef\KVo@tempa{\KVo@tempa,\CurrentOption,}%
3010       \@expandtwoargs\@removeelement\CurrentOption
3011         \@unusedoptionlist\@unusedoptionlist
3012     }%
3013   }%
3014   \edef\KVo@tempa{%
3015     \noexpand\setkeys{#1}{%
3016       \KVo@tempa\@optionlist{\@currname.\@currentx}%
3017     }%
3018   }%
3019   \KVo@tempa
3020   \AtEndOfPackage{\let\@unprocessedoptions\relax}%
3021   \let\CurrentOption\@empty
3022 }
3023 \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).

```

3024 \MT@addto@setup{%
3025 \ifMT@draft

```

We disable most of what we've just defined in the 3025 lines above if we are running in draft mode.

```

3026 \MT@warning@n1{`draft' option active.\MessageBreak
3027           Disabling all micro-typographic extensions.\MessageBreak
3028           This might lead to different line and page breaks}
3029 \MT@protrusionfalse
3030 \MT@expansionfalse
3031  $\beta$ 
3032 \MT@spacingfalse
3033 \MT@kerningfalse
3034 \MT@babelfalse
3035  $\beta$ 
3036 \let\MT@setupfont\relax

```

```

3037 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
3038 \renewcommand*{\UseMicrotypeSet[2] [] {}
3039 \renewcommand*{\SetProtrusion[3] [] {}
3040 \renewcommand*{\SetExpansion[3] [] {}
3041 <*\beta>
3042 \renewcommand*{\SetExtraSpacing[3] [] {}
3043 \renewcommand*{\SetExtraKerning[3] [] {}
3044 </\beta>
3045 \renewcommand*{\DeclareCharacterInheritance[3] [] {}
3046 \renewcommand*{\DeclareMicrotypeAlias[2] {}
3047 \renewcommand*{\LoadMicrotypeFile[1] {}
3048 \renewcommand*{\microtypesetup[1] {}
3049 \renewcommand*{\microtypecontext[1] {}
3050 \else

```

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

```

3051 \ifnum\pdfoutput<\@ne
3052 \ifMT@opt@expansion \else
3053 \MT@expansionfalse
3054 \fi
3055 \fi

```

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

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

```

3056 \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3057 \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%

```

Protrusion.

```

3058 \ifMT@protrusion
3059 \edef\MT@active@features{\MT@active@features,pr}
3060 \pdfprotrudechars\MT@pr@level
3061 \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
3062 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3063 factor: \number\MT@pr@factor\fi
3064 \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.

```

3065 \MT@ifdefined@c@TF\MT@pr@setname{%
3066 \MT@info@nl{Using protrusion set ``\MT@pr@setname'}%
3067 }{%
3068 \MT@gllet\MT@pr@setname\MT@default@pr@set
3069 \MT@info@nl{Using default protrusion set ``\MT@pr@setname'}%
3070 }
3071 \else
3072 \let\MT@protrusion\relax
3073 \MT@info@nl{No character protrusion}
3074 \fi

```

Expansion.

```

3075 \ifMT@expansion

```



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

```
3076 \ifnum\MT@stretch=\m@ne
3077 \let\MT@stretch\MT@stretch@default
3078 \fi
```

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

```
3079 \ifnum\MT@shrink=\m@ne
3080 \ifnum\MT@stretch>\z@
3081 \let\MT@shrink\MT@stretch
3082 \else
3083 \let\MT@shrink\MT@shrink@default
3084 \fi
3085 \fi
```

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

```
3086 \ifnum\MT@step=\m@ne
3087 \ifnum\MT@stretch>\MT@shrink
3088 \ifnum\MT@shrink=\z@
3089 \@tempcnta=\MT@stretch
3090 \else
3091 \@tempcnta=\MT@shrink
3092 \fi
3093 \else
3094 \ifnum\MT@stretch=\z@
3095 \@tempcnta=\MT@shrink
3096 \else
3097 \@tempcnta=\MT@stretch
3098 \fi
3099 \fi
3100 \divide\@tempcnta 5\relax
3101 \else
3102 \@tempcnta=\MT@step
3103 \ifnum\@tempcnta=\z@
3104 \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
3105 Setting it to one}
3106 \fi
3107 \fi
3108 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3109 \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 $\text{\TeX}$ ).

```
3110 \let\MT@auto\@empty
3111 \ifMT@auto
3112 \MT@requires@pdftex4{%
```

We turn off automatic expansion if output mode is DVI.

```
3113 \ifnum\pdfoutput<\@ne
3114 \ifMT@opt@auto
3115 \MT@warning@nl{%
3116 Automatic font expansion only works for PDF output.\MessageBreak
3117 However, you are creating a DVI file. I will switch.\MessageBreak
3118 automatic font expansion off and hope that expanded.\MessageBreak
3119 fonts are available}
3120 \fi
3121 \MT@autofalse
3122 \else
3123 \def\MT@auto{autoexpand}
```

```
3124 \fi
```

Also, if pdfTeX is too old.

```
3125 }{%
3126 \MT@warning@nl{%
3127     The pdftex version you are using is too old for\MessageBreak
3128     automatic font expansion. I will switch it off and\MessageBreak
3129     hope that expanded fonts are available.\MessageBreak
3130     Otherwise, install pdftex version 1.20 or newer}
3131 \MT@autofalse
3132 \def\MT@auto{1000 }
3133 }
```

No automatic expansion.

```
3134 \else
3135 \ifnum\MT@pdftex@no < 4
3136 \def\MT@auto{1000 }
3137 \fi
3138 \fi
```

Choose the appropriate macro for selected expansion.

```
3139 \ifMT@selected
3140 \let\MT@set@ex@codes\MT@set@ex@codes@s
3141 \else
3142 \let\MT@set@ex@codes\MT@set@ex@codes@n
3143 \fi
```

Filter out stretch=0, shrink=0, since it would result in an pdfTeX error.

```
3144 \ifnum\MT@stretch=\z@
3145 \ifnum\MT@shrink=\z@
3146 \MT@warning@nl{%
3147     Both the stretch and shrink limit are set to zero.\MessageBreak
3148     Disabling font expansion}
3149 \MT@expansionfalse
3150 \fi
3151 \fi
3152 \fi
3153 \ifMT@expansion
3154 \edef\MT@active@features{\MT@active@features,ex}%
3155 \pdfadjustspacing\MT@ex@level
3156 \MT@info@nl{\ifMT@auto\else Non-\fi Automatic font expansion enabled
3157     (level \number\MT@ex@level),\MessageBreak
3158     stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3159     step: \number\MT@step, \ifMT@selected\else non-\fi selected}
3160 \MT@ifdefined@c@TF\MT@ex@setname{%
3161 \MT@info@nl{Using expansion set `~\MT@ex@setname'}%
3162 }{%
3163 \MT@glet\MT@ex@setname\MT@default@ex@set
3164 \MT@info@nl{Using default expansion set `~\MT@ex@setname'}%
3165 }
```

Inside \showhyphens, font expansion should be disabled.

```
3166 \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3167 \color@begingroup\everypar{}\parfillskip\z@skip
3168 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3169 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
```

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

```
3170 \gdef\showhyphens#1{\setbox0\vbox{%
3171 \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3172 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3173 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
```

```

3174 \else
3175   \let\MT@expansion\relax
3176   \MT@info@nl{No font expansion}
3177 \fi
3178 \fi
3179 }
3180 *beta
3181 \MT@requires@pdftex6{
3182   \MT@addto@setup{%

```

### Spacing.

```

3183   \ifMT@spacing
3184     \edef\MT@active@features{\MT@active@features,sp}
3185     \pdfadjustinterwordglue\@ne
3186     \MT@info@nl{Adjustment of interword spacing enabled}
3187     \MT@ifdefined@c@TF\MT@sp@setname{%
3188       \MT@info@nl{Using spacing set `~\MT@sp@setname'}%
3189     }{%
3190       \MT@glet\MT@sp@setname\MT@default@sp@set
3191       \MT@info@nl{Using default spacing set `~\MT@sp@setname'}%
3192     }
3193   \else
3194     \let\MT@spacing\relax
3195     \MT@info@nl{No adjustment of interword spacing}
3196   \fi

```

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

```

3197   \edef\MT@active@features{\MT@active@features,kn}
3198   \ifMT@kerning
3199     \pdfprependkern\@ne
3200     \pdfappendkern\@ne
3201     \MT@info@nl{Adjustment of character kerning enabled}
3202     \MT@ifdefined@c@TF\MT@kn@setname{%
3203       \MT@info@nl{Using kerning set `~\MT@kn@setname'}%
3204     }{%
3205       \MT@glet\MT@kn@setname\MT@default@kn@set
3206       \MT@info@nl{Using default kerning set `~\MT@kn@setname'}%
3207     }
3208   \else
3209     \MT@info@nl{No adjustment of character kerning}
3210   \fi
3211   \ifnum\MT@letterspacing=\m@ne
3212     \let\MT@letterspacing\MT@letterspacing@default
3213   \fi
3214 }

```

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

```

3215 }{
3216   \MT@addto@setup{%
3217     \ifMT@spacing
3218       \MT@warning@nl{Adjustment of interword spacing only works with\MessageBreak
3219         pdftex version 1.40 or newer. Switching it off}%
3220     \else
3221       \MT@info@nl{No adjustment of interword spacing (pdftex too old)}
3222     \fi
3223     \MT@spacingfalse
3224     \let\MT@spacing\relax
3225     \ifMT@kerning
3226       \MT@warning@nl{Character kerning only works with\MessageBreak
3227         pdftex version 1.40 or newer. Switching it off}%
3228     \else

```

```

3229     \MT@info@nl{No adjustment of character kerning (pdfTeX too old)}
3230     \fi
3231     \MT@kerningfalse
3232     \let\MT@kerning\relax
3233   }
3234 }

```

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.

```

3235 \MT@requires@pdfTeX6{
3236   \AtBeginDocument{%
3237     \ifMT@spacing
3238       \ifMT@babel \else
3239         \ifnum\sfcode`. > 1500
3240           \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
3241             \MT@warning@nl{%
3242               \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3243               interword spacing will disable it. You might want\MessageBreak
3244               to add ` \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
3245               to your preamble}%
3246             }%
3247           \fi
3248         \fi
3249       \fi
3250     }
3251   }\relax
3252 }(beta)

```

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

```

3253 \MT@addto@setup{%
3254   \ifx\MT@active@features\@empty \else
3255     \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3256   \fi
3257   \MT@documenttrue
3258 }

```

`\MT@set@babel@context` Interaction with babel.

```

3259 (beta)
3260 \def\MT@set@babel@context#1{%
3261   \MT@ifdefined@n@TF{MT@babel@#1}{%
3262     \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
3263     \expandafter\MT@exp@one@n\expandafter\microtypecontext
3264     \csname MT@babel@#1\endcsname
3265   }{%
3266     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3267   }%
3268 }

```

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

```

3269 \MT@addto@setup{%
3270   \ifMT@babel
3271     \ifpackage@loaded{babel}{%
3272       \MT@info@nl{Redefining babel's language switching commands}
3273       \let\MT@orig@select@language\select@language
3274       \def\select@language#1{%
3275         \MT@orig@select@language{#1}%
3276         \MT@set@babel@context{#1}%

```

```

3277     }
3278     \let\MT@orig@foreign@language\foreign@language
3279     \def\foreign@language#1{%
3280         \MT@orig@foreign@language{#1}%
3281         \MT@set@babel@context{#1}%
3282     }

```

Disable French babel's active characters.

```

3283     \ifMT@kerning
3284     \MT@if@false
3285     \ifpackagewith{babel}{french}\MT@if@true\relax
3286     \ifpackagewith{babel}{frenchb}\MT@if@true\relax
3287     \ifpackagewith{babel}{francais}\MT@if@true\relax
3288     \ifMT@if@
3289         \MT@info@nl{Switching off French babel's active punctuation characters}%
3290         \NoAutoSpaceBeforeFDP
3291     \fi
3292 \fi

```

In case babel was loaded before microtype:

```

3293     \MT@set@babel@context\languagename
3294 }{%
3295     \MT@warning@nl{You did not load the babel package.\MessageBreak
3296         The `babel' option won't have any effect}
3297 }
3298 \fi
3299 }
3300 /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.

```

3301 \AtBeginDocument\selectfont

```

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

```

3302 \edef\MT@curr@file{\jobname.tex}

```

That was that.

```

3303 /package

```

## 14 Configuration Files

Let's now write the font configuration files.

```

3304 *config
3305

```

### 14.1 Font Sets

We first declare some sets in the main configuration file.

```

3306 *m-t
3307 %%% -----
3308 %%% FONT SETS
3309
3310 \DeclareMicrotypeSet{all}
3311 { }
3312
3313 \DeclareMicrotypeSet{allmath}
3314 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
3315

```

```

3316 \DeclareMicrotypeSet{alltext}
3317 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
3318
3319 \DeclareMicrotypeSet{basicmath}
3320 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
3321   family   = {rm*,sf*},
3322   series    = {md*},
3323   size      = {normalsize,footnotesize,small,large}
3324 }
3325
3326 \DeclareMicrotypeSet{basictext}
3327 { encoding = {OT1,T1,LY1,OT4,QX,T5},
3328   family   = {rm*,sf*},
3329   series    = {md*},
3330   size      = {normalsize,footnotesize,small,large}
3331 }
3332
3333 \DeclareMicrotypeSet{normalfont}
3334 { font = */*/*/*/ }
3335

```

The default sets.

```

3336 %%% -----
3337 %%% DEFAULT SETS
3338
3339 \DeclareMicrotypeSetDefault[protrusion]{alltext}
3340 \DeclareMicrotypeSetDefault[expansion]{basictext}
3341 <beta>
3342 \DeclareMicrotypeSetDefault[spacing]{basictext}
3343 \DeclareMicrotypeSetDefault[kerning]{alltext}
3344 </beta>
3345

```

## 14.2 Font Aliases

Fonts that are ‘the same’.

```

3346 %%% -----
3347 %%% FONT ALIASES
3348

```

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.

```

3349 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
3350 \DeclareMicrotypeAlias{aer}{cmr} % ae
3351 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
3352 \DeclareMicrotypeAlias{cmor}{cmr} % eco
3353 \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.

```

3354 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
3355 \DeclareMicrotypeAlias{qpl}{ppl} % qfonts/QuasiPalatino
3356 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
3357 \DeclareMicrotypeAlias{qtm}{ptm} % qfonts/QuasiTimes

```

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

The `eulervm` package virtually extends the Euler fonts.

```
3358 \DeclareMicrotypeAlias{eur}{eur} % Euler VM
3359 \DeclareMicrotypeAlias{zeus}{eus}
```

The `mathdesign` package provides math fonts matching Bitstream Charter and URW Garamond.

```
3360 \DeclareMicrotypeAlias{mbch}{bch} % mathdesign/Charter
3361 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

Euro symbol fonts, to save some files.

```
3362 \DeclareMicrotypeAlias{zpeus}{zpeu} % Adobe Euro sans -> serif
3363 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
3364 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
3365
```

## 14.3 Interaction with babel

Contexts that are to be set when switching to a language.

```
3366 <*\beta>
3367 %%% -----
3368 %%% INTERACTION WITH THE `babel' PACKAGE
3369
3370 \DeclareMicrotypeBabelHook
3371   {french,français}
3372   {kerning=french, spacing=}
3373
3374 \DeclareMicrotypeBabelHook
3375   {english,UKenglish,british,american,USenglish}
3376   {kerning=, spacing=nonfrench}
3377
3378 \DeclareMicrotypeBabelHook
3379   {turkish}
3380   {kerning=turkish, spacing=}
3381
3382 </\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  $\LaTeX$  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, or else in combination with the ‘inputenc’ key.

## 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 CE for O.

```
3383 </m-t>
3384 <*m-t|zpeu|mvs>
3385 %%% -----
3386 %%% CHARACTER INHERITANCE
3387
3388 </m-t|zpeu|mvs>
3389 <*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’), Æ, æ, Œ, œ.

```
3390 \DeclareCharacterInheritance
3391 { encoding = OT1 }
3392 { f = {011}, % ff
3393   i = {\i},
3394   j = {\j},
3395   o = {\0},
3396   o = {\o},
3397 }
3398
```

### 14.5.2 T1

Candidates here: 028 (‘fi’), 029 (‘fl’), 030 (‘ffi’), 031 (‘ffl’), 156 (‘IJ’ ligature, since L<sup>A</sup>T<sub>E</sub>X 2005/12/01 accessible as \IJ), 188 (‘ij’, \ij), Æ, æ, Œ, œ.

```
3399 \DeclareCharacterInheritance
3400 { encoding = T1 }
3401 { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
3402   a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
3403   C = {\`C,\'C,\^C,\~C},
3404   c = {\`c,\'c,\^c,\~c},
3405   D = {\v D,\DH},
3406   d = {\v d,\dj},
3407   E = {\`E,\'E,\^E,\"E,\k E,\v E},
3408   e = {\`e,\'e,\^e,\"e,\k e,\v e},
3409   f = {027}, % ff
3410   G = {\u G},
3411   g = {\u g},
3412   I = {\`I,\'I,\^I,\"I,\.I},
3413   i = {\`i,\'i,\^i,\"i,\i},
3414   j = {\j},
3415   L = {\L,\'L,\v L},
3416   l = {\l,\'l,\v l},
3417   N = {\`N,\~N,\v N},
3418   n = {\`n,\~n,\v n},
3419   O = {\0,\`0,\'0,\^0,\~0,\"0,\H 0},
```



```

3420 o = {\o,\`o,\'o,\^o,\~o,\"o,\H o},
3421 R = {\'R,\v R},
3422 r = {\'r,\v r},
3423 S = {\'S,\c S,\v S,\SS},
3424 s = {\'s,\c s,\v s},
3425 T = {\c T,\v T},
3426 t = {\c t,\v t},
3427 U = {\`U,\'U,\^U,\"U,\H U,\r U},
3428 u = {\`u,\'u,\^u,\"u,\H u,\r u},
3429 Y = {\'Y,\"Y},
3430 y = {\'y,\"y},
3431 Z = {\'Z,\.Z,\v Z},
3432 z = {\'z,\.z,\v z},
3433 - = {127},
3434 }
3435

```

### 14.5.3 LY1

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

```

3436 \DeclareCharacterInheritance
3437 { encoding = LY1 }
3438 { A = {\`A,\'A,\^A,\~A,\"A,\r A},
3439   a = {\`a,\'a,\^a,\~a,\"a,\r a},
3440   C = {\c C},
3441   c = {\c c},
3442   D = {\DH},
3443   E = {\`E,\'E,\^E,\"E},
3444   e = {\`e,\'e,\^e,\"e},
3445   f = {011}, % ff
3446   I = {\`I,\'I,\^I,\"I},
3447   i = {\`i,\'i,\^i,\"i,\i},
3448   L = {\L},
3449   l = {\l},
3450   N = {\~N},
3451   n = {\~n},
3452   O = {\`O,\'O,\^O,\~O,\"O,\O},
3453   o = {\`o,\'o,\^o,\~o,\"o,\o},
3454   S = {\v S},
3455   s = {\v s},
3456   U = {\`U,\'U,\^U,\"U},
3457   u = {\`u,\'u,\^u,\"u},
3458   Y = {\'Y,\"Y},
3459   y = {\'y,\"y},
3460   Z = {\v Z},
3461   z = {\v z},
3462 }
3463

```

### 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.

```

3464 \DeclareCharacterInheritance
3465 { encoding = OT4 }
3466 { A = {\k A},
3467   a = {\k a},
3468   C = {\'C},
3469   c = {\'c},
3470   E = {\k E},

```

```

3471     e = {\k e},
3472     f = {011}, % ff
3473     i = {\i},
3474     j = {\j},
3475     L = {\L},
3476     l = {\l},
3477     N = {\'N},
3478     n = {\'n},
3479     O = {\0,\'0},
3480     o = {\o,\'o},
3481     S = {\'S},
3482     s = {\'s},
3483     Z = {\'Z,\.Z},
3484     z = {\'z,\.z},
3485   }
3486

```

### 14.5.5 QX

The Central European QX encoding. Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

3487 \DeclareCharacterInheritance
3488   { encoding = QX }
3489   { A = {\^A,\'A,\^A,\~A,\"A,\k A,\AA},
3490     a = {\^a,\'a,\^a,\~a,\"a,\k a,\aa},
3491     C = {\'C,\c C},
3492     c = {\'c,\c c},
3493     D = {\DH},
3494     E = {\^E,\'E,\^E,\"E,\k E},
3495     e = {\^e,\'e,\^e,\"e,\k e},
3496     f = {011}, % ff
3497     I = {\^I,\'I,\^I,\"I,\k I},
3498     i = {\^i,\'i,\^i,\"i,\k i,\i},
3499     j = {\j},
3500     L = {\L},
3501     l = {\l},
3502     N = {\'N,\~N},
3503     n = {\'n,\~n},
3504     O = {\0,\^0,\'0,\^0,\~0,\"0},
3505     o = {\o,\^o,\'o,\^o,\~o,\"o},
3506     S = {\'S,\c S,\v S},
3507     s = {\'s,\c s,\v s},
3508     T = {\c T},
3509     t = {\c t},
3510     U = {\^U,\'U,\^U,\"U,\k U},
3511     u = {\^u,\'u,\^u,\"u,\k u},
3512     Y = {\'Y,\"Y},
3513     y = {\'y,\"y},
3514     Z = {\'Z,\.Z,\v Z},
3515     z = {\'z,\.z,\v z},
3516     . = \textellipsis
3517   }
3518

```

### 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.

```

3519 \DeclareCharacterInheritance

```

```

3520 { encoding = T5 }
3521 { A = { \^A, \'A, \-A, \h A, \d A, \^A, \u A,
3522         \~\Acircumflex, \'\Acircumflex, \-\Acircumflex, \h\Acircumflex, \d\Acircumflex,
3523         \~\Abreve, \'\Abreve, \-\Abreve, \h\Abreve, \d\Abreve},
3524       a = { \^a, \'a, \-a, \h a, \d a, \^a, \u a,
3525             \~\acircumflex, \'\acircumflex, \-\acircumflex, \h\acircumflex, \d\acircumflex,
3526             \~\abreve, \'\abreve, \-\abreve, \h\abreve, \d\abreve},
3527       D = {\DJ},
3528       d = {\dj},
3529       E = { \^E, \'E, \-E, \h E, \d E, \^E,
3530             \~\Ecircumflex, \'\Ecircumflex, \-\Ecircumflex, \h\Ecircumflex, \d\Ecircumflex},
3531       e = { \^e, \'e, \-e, \h e, \d e, \^e,
3532             \~\ecircumflex, \'\ecircumflex, \-\ecircumflex, \h\ecircumflex, \d\ecircumflex},
3533       I = { \^I, \'I, \-I, \h I, \d I},
3534       i = { \^i, \'i, \-i, \h i, \d i, \i},
3535       O = { \^O, \'O, \-O, \h O, \d O, \^O, \horn O,
3536             \~\Ocircumflex, \'\Ocircumflex, \-\Ocircumflex, \h\Ocircumflex, \d\Ocircumflex,
3537             \~\Ohorn, \'\Ohorn, \-\Ohorn, \h\Ohorn, \d\Ohorn},
3538       o = { \^o, \'o, \-o, \h o, \d o, \^o, \horn o,
3539             \~\ocircumflex, \'\ocircumflex, \-\ocircumflex, \h\ocircumflex, \d\ocircumflex,
3540             \~\ohorn, \'\ohorn, \-\ohorn, \h\ohorn, \d\ohorn},
3541       U = { \^U, \'U, \-U, \h U, \d U, \horn U,
3542             \~\Uhorn, \'\Uhorn, \-\Uhorn, \h\Uhorn, \d\Uhorn},
3543       u = { \^u, \'u, \-u, \h u, \d u, \horn u,
3544             \~\uhorn, \'\uhorn, \-\uhorn, \h\uhorn, \d\uhorn},
3545       Y = { \^Y, \'Y, \-Y, \h Y, \d Y},
3546       y = { \^y, \'y, \-y, \h y, \d y},
3547     }
3548
3549 </m-t>

```

### 14.5.7 Euro symbols

Make Euro symbols settings simpler.

```

3550 < *zpeu >
3551 \DeclareCharacterInheritance
3552 { encoding = U,
3553   family   = {zpeu,zpeus,eurosans} }
3554 { E = 128 }
3555
3556 < /zpeu >
3557 < *mvs >
3558 \DeclareCharacterInheritance
3559 { encoding = OT1,
3560   family   = mvs }
3561 { 164 = {099,100,101} } % \EURhv, \EURcr, \EURtm
3562

```

Since 2006/05/11 (that is, one week after I've added these settings, after the package had been dormant for six years), marvosym's encoding is (correctly) U instead of OT1.

```

3563 \DeclareCharacterInheritance
3564 { encoding = U,
3565   family   = mvs }
3566 { 164 = {099,100,101} }
3567
3568 < /mvs >

```

## 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).

```

3569 < *m-t >
3570 %%% -----
3571 %%% EXPANSION SETTINGS
3572
3573 \SetExpansion
3574 [ name = default ]
3575 { encoding = {OT1,OT4,QX,T1,LY1} }
3576 {
3577     A = 500,      a = 700,
3578     \AE = 500,    \ae = 700,
3579     B = 700,      b = 700,
3580     C = 700,      c = 700,
3581     D = 500,      d = 700,
3582     E = 700,      e = 700,
3583     F = 700,
3584     G = 500,      g = 700,
3585     H = 700,      h = 700,
3586     K = 700,      k = 700,
3587     M = 700,      m = 700,
3588     N = 700,      n = 700,
3589     O = 500,      o = 700,
3590     \OE = 500,    \oe = 700,
3591     P = 700,      p = 700,
3592     Q = 500,      q = 700,
3593     R = 700,
3594     S = 700,      s = 700,
3595     U = 700,      u = 700,
3596     W = 700,      w = 700,
3597     Z = 700,      z = 700,
3598     2 = 700,
3599     3 = 700,
3600     6 = 700,
3601     8 = 700,
3602     9 = 700,
3603 }
3604

```

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

```

3605 \SetExpansion
3606 [ name = T5 ]
3607 { encoding = T5 }
3608 {
3609     A = 500,      a = 700,
3610     B = 700,      b = 700,
3611     C = 700,      c = 700,
3612     D = 500,      d = 700,
3613     E = 700,      e = 700,
3614     F = 700,
3615     G = 500,      g = 700,
3616     H = 700,      h = 700,
3617     K = 700,      k = 700,
3618     M = 700,      m = 700,
3619     N = 700,      n = 700,
3620     O = 500,      o = 700,
3621     P = 700,      p = 700,
3622     Q = 500,      q = 700,
3623     R = 700,
3624     S = 700,      s = 700,

```

```

3625     U = 700,      u = 700,
3626     W = 700,      w = 700,
3627     Z = 700,      z = 700,
3628     2 = 700,
3629     3 = 700,
3630     6 = 700,
3631     8 = 700,
3632     9 = 700,
3633   }
3634
3635 /m-t

```

## 14.7 Character Protrusion

```

3636 %%% -----
3637 %%% PROTRUSION SETTINGS
3638

```

For future historians, Hàn Thế Thành's original settings (from `protcode.tex`, converted to microtype notation).

```

\SetProtrusion
[ name      = thanh ]
{ encoding = OT1 }
{
  A = {50,50},
  F = { ,50},
  J = {50, },
  K = { ,50},
  L = { ,50},
  T = {50,50},
  V = {50,50},
  W = {50,50},
  X = {50,50},
  Y = {50,50},
  k = { ,50},
  r = { ,50},
  t = { ,50},
  v = {50,50},
  w = {50,50},
  x = {50,50},
  y = {50,50},
  . = { ,700},    {,}= { ,700},
  : = { ,500},    ; = { ,500},
  ! = { ,200},    ? = { ,200},
  ( = {50, },    ) = { ,50},
  - = { ,700},
  \textendash      = { ,300},    \textemdash      = { ,200},
  \textquoteleft   = {700, },    \textquoteright  = { ,700},
  \textquotedblleft = {500, },    \textquotedblright = { ,500},
}

```

### 14.7.1 Default

The default settings always use the most moderate value.

```

3639 *cfg-t
3640 \SetProtrusion
3641 /m-t [ name      = default ]

```

We also create configuration files for the fonts Bitstream Charter (NFSS code `bch`),

```

3642 /bch [ name      = bch-default ]

```

Computer Modern Roman (cmr),

```
3643 <cmr> [ name = cmr-default ]
```

Adobe Garamond (pad, padx, padj),

```
3644 <pad> [ name = pad-default ]
```

Minion<sup>16</sup> (pmnx, pmnj),

```
3645 <pmn> [ name = pmnj-default ]
```

Palatino (ppl, pplx, pplj),

```
3646 <ppl> [ name = ppl-default ]
```

Times (ptm, ptmx, ptmj),

```
3647 <ptm> [ name = ptm-default ]
```

and URW Garamond (ugm).

```
3648 <ugm> [ name = ugm-default ]
3649 <m-t> { encoding = OT1 }
3650 <cmr> { }
3651 <bch|pad|pmn|ugm> { encoding = OT1,
3652 <ppl|ptm> { encoding = {OT1,OT4},
3653 <bch> family = bch }
3654 <pad> family = {pad,padx,padj} }
3655 <pmn> family = pmnj }
3656 <ppl> family = {ppl,pplx,pplj} }
3657 <ptm> family = {ptm,ptmx,ptmj} }
3658 <ugm> family = ugm }
3659 {
3660 <m-t|bch|cmr|pad|pmn|ppl|ptm> A = {50,50},
3661 <ugm> A = {50,100},
3662 <m-t|pad|ptm> \AE = {50, },
3663 <ugm> \AE = {150,50},
3664 <ugm> B = { ,50},
3665 <bch|pad|pmn|ugm> C = {50, },
3666 <bch|pad|pmn> D = { ,50},
3667 <ugm> D = { ,70},
3668 <ugm> E = { ,50},
3669 <m-t|bch|cmr|pad|pmn|ptm> F = { ,50},
3670 <ugm> F = { ,70},
3671 <bch|pad|pmn> G = {50, },
3672 <ugm> G = {50,50},
3673 <m-t|cmr|pad|pmn|ppl|ptm|ugm> J = {50, },
3674 <bch> J = {100, },
3675 K = { ,50},
3676 <m-t|bch|cmr|pad|pmn|ppl> L = { ,50},
3677 <ptm> L = { ,80},
3678 <ugm> L = { ,120},
3679 <bch|pad|pmn|ugm> O = {50,50},
3680 <pad|pmn> \OE = {50, },
3681 <ugm> \OE = {50,50},
3682 <ugm> P = { ,50},
3683 <bch|pad|pmn> Q = {50,70},
3684 <ugm> Q = {50,50},
3685 <bch> R = { ,50},
3686 <ugm> R = { ,70},
3687 <m-t|bch|cmr|pad|pmn|ppl|ptm> T = {50,50},
3688 <ugm> T = {70,70},
3689 <m-t|bch|cmr|pad|pmn|ppl|ptm> V = {50,50},
3690 <ugm> V = {70,70},
3691 <m-t|bch|cmr|pad|pmn|ppl|ptm> W = {50,50},
3692 <ugm> W = {70,70},
```

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

```

3693 <m-t|bch|cmr|pad|pmn|ppl|ptm>    X = {50,50},
3694 <ugm>    X = {50,70},
3695 <m-t|bch|cmr|pad|pmn|ppl>    Y = {50,50},
3696 <ptm|ugm>    Y = {80,80},
3697 <ugm>    Z = {50,50},
3698 <m-t|bch|cmr|pad|pmn|ppl|ptm>    k = { ,50},
3699 <ugm>    k = { ,70},
3700 <pmn>    l = { ,-50},
3701 <pad|ppl>    p = {50,50},
3702 <ugm>    p = { ,50},
3703 <pad|ppl>    q = {50, },
3704    r = { ,50},
3705 <cmr|pad|pmn>    t = { ,70},
3706 <bch>    t = { ,50},
3707 <ugm>    t = { ,100},
3708 <m-t|bch|cmr|pad|pmn|ppl|ptm>    v = {50,50},
3709 <ugm>    v = {50,70},
3710 <m-t|bch|cmr|pad|pmn|ppl|ptm>    w = {50,50},
3711 <ugm>    w = {50,70},
3712    x = {50,50},
3713 <m-t|bch|pad|pmn>    y = { ,50},
3714 <cmr|ppl|ptm>    y = {50,70},
3715 <ugm>    y = { ,70},

3716 <cmr>    0 = { ,50},
3717 <m-t>    1 = {50,50},
3718 <bch|pad|ptm|ugm>    1 = {150,150},
3719 <cmr>    1 = {100,200},
3720 <pmn>    1 = { ,50},
3721 <ppl>    1 = {100,100},
3722 <bch|cmr|pad|ugm>    2 = {50,50},
3723 <cmr|pad|ugm>    3 = {50,50},
3724 <bch|pmn>    3 = {50, },
3725 <m-t|pad>    4 = {50,50},
3726 <bch>    4 = {100,50},
3727 <cmr|ugm>    4 = {70,70},
3728 <pmn>    4 = {50, },
3729 <ptm>    4 = {70, },
3730 <cmr>    5 = { ,50},
3731 <pad>    5 = {50,50},
3732 <bch>    6 = {50, },
3733 <cmr>    6 = { ,50},
3734 <pad>    6 = {50,50},
3735 <m-t>    7 = {50,50},
3736 <bch|pad|pmn|ugm>    7 = {50,80},
3737 <cmr|ptm>    7 = {50,100},
3738 <ppl>    7 = { ,50},
3739 <cmr>    8 = { ,50},
3740 <bch|pad>    9 = {50,50},
3741 <cmr>    9 = { ,50},
3742 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    . = { ,700},
3743 <bch>    . = { ,600},
3744    {,}= { ,500},
3745 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    : = { ,500},
3746 <bch>    : = { ,400},
3747 <m-t|bch|pad|pmn|ptm>    ; = { ,300},
3748 <cmr|ppl>    ; = { ,500},
3749 <ugm>    ; = { ,400},
3750    ! = { ,100},
3751 <m-t|pad|pmn|ptm>    ? = { ,100},
3752 <bch|cmr|ppl|ugm>    ? = { ,200},
3753 <pmn>    " = {300,300},
3754 <m-t|bch|cmr|pad|pmn|ppl>    @ = {50,50},

```

```

3755 <ptm> @ = {100,100},
3756 <m-t|bch|cmr|pad|pmn|ppl|ptm> ~ = {200,250},
3757 <ugm> ~ = {300,350},
3758 <pad|ppl|ptm> & = {50,100},
3759 <ugm> & = { ,100},
3760 <m-t|cmr|pad|pmn> \% = {50,50},
3761 <bch> \% = { ,50},
3762 <ppl|ptm> \% = {100,100},
3763 <ugm> \% = {50,100},
3764 <m-t|ppl|ptm|ugm> * = {200,200},
3765 <bch|pmn> * = {200,300},
3766 <cmr|pad> * = {300,300},
3767 <m-t|cmr|ppl|ptm> + = {250,250},
3768 <bch> + = {150,250},
3769 <pad> + = {300,300},
3770 <pmn> + = {150,200},
3771 <ugm> + = {250,300},
3772 <ugm> {=} = {200,200},
3773 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
3774 <bch|ugm> ( = {200, }, ) = { ,200},
3775 <cmr|ppl> ( = {100, }, ) = { ,300},
3776 <bch|pmn> [ = {100, }, ] = { ,100},

3777 <m-t|pad|pmn|ptm> / = {100,200},
3778 <bch> / = { ,200},
3779 <cmr|ppl> / = {200,300},
3780 <ugm> / = {100,300},
3781 <m-t|ptm> - = {500,500},
3782 <bch|cmr|ppl> - = {400,500},
3783 <pad> - = {300,500},
3784 <pmn> - = {200,400},
3785 <ugm> - = {500,600},
3786 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
3787 <bch> \textendash = {200,300}, \textendash = {150,250},
3788 <cmr> \textendash = {400,300}, \textendash = {300,200},
3789 <pad|ppl|ptm> \textendash = {300,300}, \textendash = {200,200},
3790 <ugm> \textendash = {250,300}, \textendash = {250,250},

```

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

```

3791 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
3792 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
3793 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
3794 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
3795 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
3796 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300},
3797 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600},
3798 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400},
3799 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400},
3800 }
3801

```

Greek uppercase letters are in OT1 encoding only.

```

3802 <*cmr>
3803 \SetProtrusion
3804 [ name = cmr-OT1,
3805   load = cmr-default ]
3806 { encoding = {OT1,OT4},
3807   family = cmr }
3808 {
3809   \AE = { 50, },
3810   "00 = { ,150}, % \Gamma

```



```

3811 "01 = {100,100}, % \Delta
3812 "02 = { 50, 50}, % \Theta
3813 "03 = {100,100}, % \Lambda
3814 "06 = { 50, 50}, % \Sigma
3815 "07 = {100,100}, % \Upsilon
3816 "08 = { 50, 50}, % \Phi
3817 "09 = { 50, 50}, % \Psi

```

Remaining slots can be found in the source file.

```

3818 }
3819
3820 </cmr>

```

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

```

3821 \SetProtrusion
3822 <m-t> [ name = T1-default,
3823 <bch> [ name = bch-T1,
3824 <cmr> [ name = cmr-T1,
3825 <pad> [ name = pad-T1,
3826 <pmn> [ name = pmnj-T1,
3827 <ppl> [ name = ppl-T1,
3828 <ptm> [ name = ptm-T1,
3829 <ugm> [ name = ugm-T1,
3830 <m-t> load = default ]
3831 <bch> load = bch-default ]
3832 <cmr> load = cmr-default ]
3833 <pad> load = pad-default ]
3834 <pmn> load = pmnj-default ]
3835 <ppl> load = ppl-default ]
3836 <ptm> load = ptm-default ]
3837 <ugm> load = ugm-default ]
3838 <m-t> { encoding = {T1,LY1} }
3839 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
3840 <ptm|ugm> { encoding = {T1},
3841 <bch> family = bch }
3842 <cmr> family = cmr }
3843 <pad> family = {pad,padx,padj} }
3844 <pmn> family = pmnj }
3845 <ppl> family = {ppl,pplx,pplj} }
3846 <ptm> family = {ptm,ptmx,ptmj} }
3847 <ugm> family = ugm }
3848 {
3849 <cmr> \AE = {50, },
3850 <bch> \OE = {50, },
3851 <pmn> \TH = { ,50},
3852 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
3853 <cmr> _ = {200,200},
3854 <ugm> _ = {100,200},
3855 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
3856 <bch> \textbackslash = {150,200},
3857 <cmr|ppl> \textbackslash = {200,300},
3858 <ugm> \textbackslash = {100,300},
3859 <ugm> \textbar = {200,200},
3860 <cmr> \textquotedblleft = {200,600},
3861 <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.

```

3862 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3863 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
3864 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},

```

```

3865 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
3866 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
3867 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
3868 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
3869 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
3870 <pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
3871 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
3872 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
3873 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
3874 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
3875 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
3876 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200},
3877 <pmn> \textless = {100, }, \textgreater = { ,100},
3878 <pmn> \textvisiblespace = {100,100}, % not in LY1
3879 }
3880

```

The lmodern fonts, on the other hand, restore the original kerning from the OT1 fonts, and so do we. Silly, isn't it?

```

3881 <*cmr>
3882 \SetProtrusion
3883 [ name = lmr-T1,
3884 load = cmr-T1 ]
3885 { encoding = {T1,LY1},
3886 family = lmr }
3887 {
3888 \textquotedblleft = {500,300},
3889 \quotedblbase = {500,300},
3890 }
3891
3892 </cmr>

```

Settings for the QX encoding (generic and Times). It also includes some glyphs otherwise in TS1.

```

3893 <*m-t|ptm>
3894 \SetProtrusion
3895 <m-t> [ name = QX-default,
3896 <ptm> [ name = ptm-QX,
3897 <m-t> load = default ]
3898 <ptm> load = ptm-default ]
3899 <m-t> { encoding = QX }
3900 <ptm> { encoding = QX,
3901 <ptm> family = {ptm,ptmx,ptmj} }
3902 {
3903 <ptm> * = {200,200},
3904 {=} = {100,100},
3905 \textunderscore = {100,100},
3906 \textbackslash = {100,200},
3907 \quotedblbase = {400,400},
3908 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
3909 <ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
3910 \textexclamdown = {100, }, \textquestiondown = {100, },
3911 <m-t> \textbraceleft = {400,200}, \textbraceright = {200,400},
3912 <ptm> \textbraceleft = {200,200}, \textbraceright = {200,300},
3913 \textless = {200,100}, \textgreater = {100,200},
3914 \textminus = {200,200}, \textdegree = {300,300},
3915 <m-t> \copyright = {100,100}, \textregistered = {100,100},
3916 <ptm> \copyright = {100,150}, \textregistered = {100,150},
3917 <ptm> \textxgeq = { ,100}, \textxleq = {100, },
3918 <ptm> \textalpha = { , 50}, \textDelta = { 70, 70},
3919 <ptm> \textpi = { 50, 80}, \textSigma = { , 70},

```

```

3920 <ptm> \textmu = { , 80}, \texteuro = { 50, 50},
3921 <ptm> \textellipsis = {150,200}, \textasciitilde = { 80, 80},
3922 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
3923 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
3924 <ptm> \textdiv = { 50,150}, \textsection = { 80, 80},
3925 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
3926 <ptm> \textbullet = {150,150}, \textperiodcentered = {300,300},
3927 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
3928 <ptm> \textperthousand = { ,50},
3929 }
3930
3931 </m-t|ptm>

```

T5 is based on OT1; it shares some but not all extra characters of T1. All accented characters are already taken care of by the inheritance list.

```

3932 <*cmr|bch>
3933 \SetProtrusion
3934 <cmr> [ name = cmr-T5,
3935 <cmr> load = cmr-default ]
3936 <bch> [ name = bch-T5,
3937 <bch> load = bch-default ]
3938 { encoding = T5,
3939 <cmr> family = cmr }
3940 <bch> family = bch }
3941 {
3942 <bch> _ = {100,100},
3943 <bch> \textbackslash = {150,200},
3944 <cmr> \textbackslash = {200,300},
3945 <cmr> \textquotedblleft = {200,600},
3946 <cmr> \textquotedbl = {300,300},
3947 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
3948 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3949 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
3950 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
3951 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},
3952 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
3953 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
3954 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
3955 \textless = {200,100}, \textgreater = {100,200},
3956 }
3957
3958 </cmr|bch>
3959 <*pmn>
3960 \SetProtrusion
3961 [ name = pmnx-OT1,
3962 load = pmnj-default ]
3963 { encoding = OT1,
3964 family = pmnx }
3965 {
3966 1 = {230,180},
3967 }
3968
3969 \SetProtrusion
3970 [ name = pmnx-T1,
3971 load = pmnj-T1 ]
3972 { encoding = {T1,LV1},
3973 family = pmnx }
3974 {
3975 1 = {230,180},
3976 }
3977
3978 </pmn>

```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```

3979 <*ptm>
3980 \SetProtrusion
3981 [ name      = ptm-LY1,
3982   load      = ptm-T1 ]
3983 { encoding = LY1,
3984   family   = {ptm,ptmx,ptmj} }
3985 {
3986   -                      = {100,100},
3987   \texttrademark         = {100,100},
3988   \textregistered        = {100,100},
3989   \textcopyright         = {100,100},
3990   \textdegree            = {300,300},
3991   \textminus             = {200,200},
3992   \textellipsis          = {150,200},
3993   \texteuro              = {  ,  }, % ?
3994   \textcent              = {100,100},
3995   \textquotesingle       = {500,500},
3996   \textflorin            = { 50, 70},
3997   \textdagger            = {150,150},
3998   \textdaggerdbl         = {100,100},
3999   \textperthousand       = {  , 50},
4000   \textbullet            = {150,150},
4001   \textonesuperior       = {100,100},
4002   \texttwosuperior       = { 50, 50},
4003   \textthreesuperior     = { 50, 50},
4004   \textperiodcentered    = {300,300},
4005   \textplusminus         = { 50, 80},
4006   \textmultiply          = {100,100},
4007   \textdivide            = { 50,150},

```

Remaining slots in the source file.

```

4008   }
4009
4010 </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.

```

4011 \SetProtrusion
4012 <m-t> [ name      = OT1-it ]
4013 <bch> [ name      = bch-it ]
4014 <cmr> [ name      = cmr-it ]
4015 <pad> [ name      = pad-it ]
4016 <pmn> [ name      = pmnj-it ]
4017 <ppl> [ name      = ppl-it ]
4018 <ptm> [ name      = ptm-it ]
4019 <ugm> [ name      = ugm-it ]
4020 <m-t|bch|pad|pmn|ugm> { encoding = OT1,
4021 <ppl|ptm> { encoding = {OT1,OT4},
4022 <bch>      family   = bch,
4023 <pad>      family   = {pad,padx,padj},
4024 <pmn>      family   = pmnj,
4025 <ppl>      family   = {ppl,pplx,pplj},
4026 <ptm>      family   = {ptm,ptmx,ptmj},
4027 <ugm>      family   = ugm,
4028 <!(cmr|ugm)> shape    = {it,sl} }

```

```

4029 <ugm>      shape    = it  }
4030 <cmr>      {  }
4031 {
4032 <cmr|ptm>    A = {100,50},
4033 <pad|pmn>    A = {50,  },
4034 <ugm>        A = {  ,150},
4035 <ppl>        A = {50,50},
4036 <ptm>        \AE = {100,  },
4037 <pad|ppl>    \AE = {50,  },
4038 <pmn>        \AE = {  ,-50},
4039 <cmr|pad|ppl|ptm> B = {50,  },
4040 <pmn>        B = {20,-50},
4041 <bch|ppl|ptm|ugm> C = {50,  },
4042 <cmr|pad>    C = {100,  },
4043 <pmn>        C = {50,-50},
4044 <cmr|pad|ppl|ptm> D = {50,50},
4045 <pmn>        D = {20,  },
4046 <cmr|pad|ppl|ptm> E = {50,  },
4047 <pmn>        E = {20,-50},
4048 <cmr|pad|ptm> F = {100,  },
4049 <pmn>        F = {10,  },
4050 <ppl>        F = {50,  },
4051 <bch|ppl|ptm|ugm> G = {50,  },
4052 <cmr|pad>    G = {100,  },
4053 <pmn>        G = {50,-50},
4054 <cmr|pad|ppl|ptm> H = {50,  },
4055 <cmr|pad|ptm> I = {50,  },
4056 <pmn>        I = {20,-50},
4057 <cmr|ptm>    J = {100,  },
4058 <pad>        J = {50,  },
4059 <pmn>        J = {20,  },
4060 <cmr|pad|ppl|ptm> K = {50,  },
4061 <pmn>        K = {20,  },
4062 <cmr|pad|ppl|ptm> L = {50,  },
4063 <pmn>        L = {20,50},
4064 <ugm>        L = {  ,100},
4065 <cmr|ptm>    M = {50,  },
4066 <pmn>        M = {  ,-30},
4067 <cmr|ptm>    N = {50,  },
4068 <pmn>        N = {  ,-30},
4069 <bch|pmn|ppl|ptm> O = {50,  },
4070 <cmr|pad>    O = {100,  },
4071 <ugm>        O = {70,50},
4072 <pmn|ppl|ptm> \OE = {50,  },
4073 <pad>        \OE = {100,  },
4074 <cmr|pad|ppl|ptm> P = {50,  },
4075 <pmn>        P = {20,-50},
4076 <bch|pmn|ppl|ptm> Q = {50,  },
4077 <cmr|pad>    Q = {100,  },
4078 <ugm>        Q = {70,50},
4079 <cmr|pad|ppl|ptm> R = {50,  },
4080 <pmn>        R = {20,  },
4081 <bch|cmr|pad|ppl|ptm> S = {50,  },
4082 <pmn>        S = {20,-30},
4083 <bch|cmr|pad|ppl|ptm> $ = {50,  },
4084 <pmn>        $ = {20,-30},
4085 <bch|pmn|ugm> T = {70,  },
4086 <cmr|pad|ppl|ptm> T = {100,  },
4087 <cmr|pad|ppl|ptm> U = {50,  },
4088 <pmn>        U = {50,-50},
4089 <cmr|pad|pmn|ugm> V = {100,  },
4090 <ppl|ptm>    V = {100,50},
4091 <cmr|pad|pmn|ugm> W = {100,  },

```

```

4092 <ppl>      W = {50, },
4093 <ptm>      W = {100,50},
4094 <cmr|ppl|ptm> X = {50, },
4095 <cmr|ptm>   Y = {100, },
4096 <pmn>      Y = {50, },
4097 <ppl>      Y = {100,50},
4098 <pmn>      Z = { , -50},
4099 <pmn>      d = { , -50},
4100 <pad|pmn>  f = { , -100},
4101 <pmn>      i = { , -30},
4102 <pmn>      j = { , -30},
4103 <pmn>      l = { , -100},
4104 <bch>      o = {50,50},
4105 <bch>      p = { , 50},
4106 <pmn>      p = {-50, },
4107 <bch>      q = {50, },
4108 <pmn>      r = { , 50},
4109 <bch>      t = { , 50},
4110 <pmn|ugm>  v = {50, },
4111 <bch>      w = { , 50},
4112 <pmn|ugm>  w = {50, },
4113 <bch>      y = { , 50},
4114 <cmr>      0 = {100, },
4115 <bch|ptm>  1 = {150,100},
4116 <cmr>      1 = {200,50},
4117 <pad>      1 = {150, },
4118 <pmn>      1 = {50, },
4119 <ppl>      1 = {100, },
4120 <ugm>      1 = {150,150},
4121 <cmr>      2 = {100,-100},
4122 <pad|ppl|ptm> 2 = {50, },
4123 <pmn>      2 = {-50, },
4124 <bch>      3 = {50, },
4125 <cmr>      3 = {100,-100},
4126 <pmn>      3 = {-100, },
4127 <ptm>      3 = {100,50},
4128 <bch>      4 = {100, },
4129 <cmr|pad>  4 = {150, },
4130 <ppl|ptm>  4 = {50, },
4131 <cmr>      5 = {100, },
4132 <ptm>      5 = {50, },
4133 <bch>      6 = {50, },
4134 <cmr>      6 = {100, },
4135 <bch|pad|ptm> 7 = {100, },
4136 <cmr>      7 = {200,-150},
4137 <pmn>      7 = {20, },
4138 <ppl>      7 = {50, },
4139 <cmr>      8 = {50,-50},
4140 <cmr>      9 = {100,-100},
4141 <m-t|cmr|pad|pmn|ppl> . = { , 500},
4142 <bch|ptm|ugm> . = { , 700},
4143 <m-t|cmr|pad|pmn|ppl> {,}= { , 500},
4144 <bch|ugm>   {,}= { , 600},
4145 <ptm>      {,}= { , 700},
4146 <m-t|cmr|pad|ppl>   : = { , 300},
4147 <bch|ugm>      : = { , 400},
4148 <pmn>      : = { , 200},
4149 <ptm>      : = { , 500},
4150 <m-t|cmr|pad|ppl>   ; = { , 300},
4151 <bch|ugm>      ; = { , 400},
4152 <pmn>      ; = { , 200},
4153 <ptm>      ; = { , 500},
4154 <ptm>      ! = { , 100},

```

```

4155 <bch>      ? = { ,200},
4156 <ptm>      ? = { ,100},
4157 <ppl>      ? = { ,300},
4158 <pmn>      " = {400,200},
4159 <m-t|pad|pmn|ppl|ptm>  & = {50,50},
4160 <bch>      & = { ,80},
4161 <cmr>      & = {100,50},
4162 <ugm>      & = {50,100},
4163 <m-t|cmr|pad|pmn>      \% = {100, },
4164 <bch>      \% = {50,50},
4165 <ppl|ptm>      \% = {100,100},
4166 <ugm>      \% = {100,50},
4167 <m-t|pmn|ppl>      * = {200,200},
4168 <bch>      * = {300,200},
4169 <cmr>      * = {400,100},
4170 <pad>      * = {500,100},
4171 <ptm|ugm>      * = {400,200},
4172 <m-t|cmr|pmn|ppl>      + = {150,200},
4173 <bch|ugm>      + = {250,250},
4174 <pad|ptm>      + = {250,200},
4175 <m-t|pad|pmn|ppl>      @ = {50,50},
4176 <bch>      @ = {80,50},
4177 <cmr>      @ = {200,50},
4178 <ptm>      @ = {150,150},
4179 <m-t|bch|ugm>      ~ = {150,150},
4180 <cmr|pad|pmn|ppl|ptm>      ~ = {200,150},
4181 <ugm>      {=} = {200,200},
4182      ( = {200, }, ) = { ,200},
4183 <m-t|cmr|pad|ppl|ptm|ugm>      / = {100,200},
4184 <bch>      / = { ,150},
4185 <pmn>      / = {100,150},
4186 <m-t>      - = {300,300},
4187 <bch|pad>      - = {300,400},
4188 <pmn>      - = {200,300},
4189 <cmr>      - = {500,300},
4190 <ppl>      - = {300,500},
4191 <ptm>      - = {500,500},
4192 <ugm>      - = {400,700},
4193 <m-t|pmn>      \textendash = {200,200}, \textendash = {150,150},
4194 <bch>      \textendash = {200,300}, \textendash = {150,200},
4195 <cmr>      \textendash = {500,300}, \textendash = {400,200},
4196 <pad|ppl|ptm|ugm>      \textendash = {300,300}, \textendash = {200,200},
4197 <m-t|bch|pmn|ugm>      \textquoteleft = {400,200}, \textquoteright = {400,200},
4198 <cmr|pad>      \textquoteleft = {800,200}, \textquoteright = {800,200},
4199 <ppl>      \textquoteleft = {700,400}, \textquoteright = {700,400},
4200 <ptm>      \textquoteleft = {800,500}, \textquoteright = {800,500},
4201 <m-t|bch|pmn>      \textquotedblleft = {400,200}, \textquotedblright = {400,200},
4202 <cmr>      \textquotedblleft = {700,100}, \textquotedblright = {500,300},
4203 <pad>      \textquotedblleft = {700,200}, \textquotedblright = {700,200},
4204 <ppl>      \textquotedblleft = {500,300}, \textquotedblright = {500,300},
4205 <ptm>      \textquotedblleft = {700,400}, \textquotedblright = {700,400},
4206 <ugm>      \textquotedblleft = {600,200}, \textquotedblright = {600,200},
4207   }
4208
4209 <*cmr>
4210 \SetProtrusion
4211   [ name      = cmr-it-OT1,
4212     load      = cmr-it  ]
4213   { encoding = {OT1,OT4},
4214     family   = cmr,
4215     shape    = it      }
4216   {
4217     \AE = {100, },

```

```

4218 \OE = {100, },
4219 "00 = {200,150}, % \Gamma
4220 "01 = {150,100}, % \Delta
4221 "02 = {150, 50}, % \Theta
4222 "03 = {150, 50}, % \Lambda
4223 "04 = {100,100}, % \Xi
4224 "05 = {100,100}, % \Pi
4225 "06 = {100, 50}, % \Sigma
4226 "07 = {200,150}, % \Upsilon
4227 "08 = {150, 50}, % \Phi
4228 "09 = {150,100}, % \Psi
4229 "0A = { 50, 50}, % \Omega
4230 }
4231
4232 </cmr>
4233 \SetProtrusion
4234 <m-t> [ name = T1-it-default,
4235 <bch> [ name = bch-it-T1,
4236 <cmr> [ name = cmr-it-T1,
4237 <pad> [ name = pad-it-T1,
4238 <pmn> [ name = pmnj-it-T1,
4239 <ppl> [ name = ppl-it-T1,
4240 <ptm> [ name = ptm-it-T1,
4241 <ugm> [ name = ugm-it-T1,
4242 <m-t> load = OT1-it ]
4243 <bch> load = bch-it ]
4244 <cmr> load = cmr-it ]
4245 <pmn> load = pmnj-it ]
4246 <pad> load = pad-it ]
4247 <ppl> load = ppl-it ]
4248 <ptm> load = ptm-it ]
4249 <ugm> load = ugm-it ]
4250 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,Ly1},
4251 <ptm|ugm> { encoding = {T1},
4252 <bch> family = bch,
4253 <cmr> family = cmr,
4254 <pmn> family = pmnj,
4255 <pad> family = {pad,padx,padj},
4256 <ppl> family = {ppl,pplx,pplj},
4257 <ptm> family = {ptm,ptmx,ptmj},
4258 <ugm> family = ugm,
4259 <!(cmr|ugm)> shape = {it,sl} }
4260 <cmr|ugm> shape = it }
4261 {
4262 <m-t|bch|pmn> _ = { ,100},
4263 <cmr|ugm> _ = {100,200},
4264 <pad|ppl|ptm> _ = {100,100},
4265 <cmr> \AE = {100, },
4266 <bch> \OE = { 50, },
4267 <cmr> \OE = {100, },
4268 <pmn> 031 = { , -100}, % ffl
4269 <cmr|ptm> 156 = {100, }, % IJ
4270 <pad> 156 = {50, }, % IJ
4271 <pmn> 156 = {20, }, % IJ
4272 <pmn> 188 = { , -30}, % ij
4273 <pmn> \v t = { ,100},
4274 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
4275 <cmr|ugm> \textbackslash = {300,300},
4276 <bch> \textbackslash = {150,150},
4277 <pmn> \textbackslash = {100,150},
4278 <ugm> \textbar = {200,200},
4279 <cmr> \textquotedblleft = {500,300},
4280 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},

```



```

4281 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4282 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4283 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
4284 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
4285 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4286 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4287 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4288 <pad> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
4289 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4290 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
4291 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
4292 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4293 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
4294 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
4295 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4296 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
4297 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4298 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
4299 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
4300 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
4301 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
4302 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
4303 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100},
4304 <pmn> \textvisiblespace = {100,100},
4305 }
4306
4307 <*m-t|ptm>
4308 \SetProtrusion
4309 <m-t> [ name = QX-it-default,
4310 <ptm> [ name = ptm-it-QX,
4311 <m-t> load = OT1-it ]
4312 <ptm> load = ptm-it ]
4313 { encoding = {QX},
4314 <ptm> family = {ptm,ptmx,ptmj},
4315 shape = {it,sl} }
4316 {
4317 <ptm> 009 = { , 50}, % fk
4318 {=} = {100,100},
4319 <m-t> \textunderscore = {100,100},
4320 <ptm> \textunderscore = {100,150},
4321 \textbackslash = {100,200},
4322 \quotedblbase = {300,400},
4323 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
4324 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
4325 \textexclamdown = {200, }, \textquestiondown = {200, },
4326 \textbraceleft = {200,100}, \textbraceright = {200,200},
4327 \textless = {100,100}, \textgreater = {100,100},
4328 \textminus = {200,200}, \textdegree = {300,150},
4329 <m-t> \copyright = {100,100}, \textregistered = {100,100},
4330 <ptm> \textregistered = {100,150}, \copyright = {100,150},
4331 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
4332 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
4333 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
4334 <ptm> \textquotelleft = {500,400}, \textquoteright = {500,400},
4335 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
4336 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
4337 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
4338 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
4339 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
4340 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
4341 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
4342 <ptm> \textperthousand = { ,50},
4343 }

```

```

4344
4345 </m-t|ptm>
4346 <*cmr|bch>
4347 \SetProtrusion
4348 <cmr> [ name = cmr-it-T5,
4349 <cmr>   load = cmr-it ]
4350 <bch> [ name = bch-it-T5,
4351 <bch>   load = bch-it ]
4352 { encoding = T5,
4353 <bch>   family = bch,
4354 <cmr>   family = cmr,
4355   shape = it }
4356 {
4357 <bch>   _ = { ,100},
4358 <cmr>   _ = {100,200},
4359 <bch>   \textbackslash = {150,150},
4360 <cmr>   \textbackslash = {300,300},
4361 <bch>   \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4362 <cmr>   \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4363 <bch>   \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4364 <cmr>   \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4365 <bch>   \guillemotleft = {200,300}, \guillemotright = {150,400},
4366 <cmr>   \guillemotleft = {400,100}, \guillemotright = {200,300},
4367 <bch>   \textbraceleft = {200, }, \textbraceright = { ,200},
4368 <cmr>   \textbraceleft = {400,100}, \textbraceright = {200,200},
4369 <bch>   \textless = {100, }, \textgreater = { ,100},
4370 <cmr>   \textless = {300,100}, \textgreater = {200,100},
4371 }
4372
4373 </cmr|bch>

```

Slanted is very similar to italic.

```

4374 <*cmr>
4375 \SetProtrusion
4376 [ name = cmr-sl,
4377   load = cmr-it-OT1 ]
4378 { encoding = {OT1,OT4},
4379   family = cmr,
4380   shape = sl }
4381 {
4382   L = { ,50},
4383   f = { , -50},
4384   - = {300, },
4385   \textendash = {400, }, \textemdash = {300, },
4386 }
4387
4388 \SetProtrusion
4389 [ name = cmr-sl-T1,
4390   load = cmr-it-T1 ]
4391 { encoding = {T1,LY1},
4392   family = cmr,
4393   shape = sl }
4394 {
4395   L = { ,50},
4396   f = { , -50},
4397   - = {300, },
4398   \textendash = {400, }, \textemdash = {300, },
4399 }
4400
4401 \SetProtrusion
4402 [ name = cmr-sl-T5,
4403   load = cmr-it-T5 ]
4404 { encoding = T5,

```

```

4405     family    = cmr,
4406     shape      = sl }
4407   {
4408     L = { ,50},
4409     f = { ,-50},
4410     - = {300, },
4411     \textendash = {400, }, \textendash = {300, },
4412   }
4413
4414 \SetProtrusion
4415   [ name      = lmr-it-T1,
4416     load      = cmr-it-T1 ]
4417   { encoding = {T1,LY1},
4418     family    = lmr,
4419     shape      = {it,sl} }
4420   {
4421     \textquotedblleft = {700,100},
4422     \quotedblbase      = {600,300},
4423   }
4424

```

Oldstyle numerals are slightly different.

```

4425 \SetProtrusion
4426   [ name = cmr(oldstyle)-it,
4427     load = cmr-it-T1 ]
4428   { encoding = T1,
4429     family    = {hfor,cmor},
4430     shape      = {it,sl} }
4431   {
4432     1 = {250, 50},
4433     2 = {150,-100},
4434     3 = {100,-50},
4435     4 = {150,150},
4436     6 = {200, },
4437     7 = {200, 50},
4438     8 = {150,-50},
4439     9 = {100, 50},
4440   }
4441
4442 /cmr
4443 *pmn
4444 \SetProtrusion
4445   [ name      = pmnx-it,
4446     load      = pmnj-it ]
4447   { encoding = OT1,
4448     family    = pmnx,
4449     shape      = {it,sl} }
4450   {
4451     1 = {100,150},
4452   }
4453
4454 \SetProtrusion
4455   [ name      = pmnx-it-T1,
4456     load      = pmnj-it-T1 ]
4457   { encoding = {T1,LY1},
4458     family    = pmnx,
4459     shape      = {it,sl} }
4460   {
4461     1 = {100,150},
4462   }
4463
4464 /pmn
4465 *ptm

```

```

4466 \SetProtrusion
4467 [ name      = ptm-it-LY1,
4468   load      = ptm-it-T1 ]
4469 { encoding = {LY1},
4470   family   = {ptm,ptmx,ptmj},
4471   shape     = {it,sl} }
4472 {
4473   _                = {100,100},
4474   \texttrademark   = {100,100},
4475   \textregistered  = {100,100},
4476   \textcopyright   = {100,100},
4477   \textdegree      = {300,100},
4478   \textminus       = {200,200},
4479   \textellipsis    = {100,200},
4480   \texteuro        = { , },
4481   \textcent        = {100,100},
4482   \textquotesingle = {500, },
4483   \textflorin      = {100, 70},
4484   \textdagger       = {150,150},
4485   \textdaggerdbl    = {100,100},
4486   \textbullet       = {150,150},
4487   \textonesuperior = {150,100},
4488   \texttwosuperior  = {150, 50},
4489   \textthreesuperior = {150, 50},
4490   \textparagraph    = {100, },
4491   \textperiodcentered = {500,300},
4492   \textonequarter   = { 50, },
4493   \textonehalf       = { 50, },
4494   \textplusminus     = {100,100},
4495   \textmultiply      = {150,150},
4496   \textdivide        = {150,150},
4497 }
4498
4499 </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).

```

4500 <!*ugm>
4501 \SetProtrusion
4502 <m-t> [ name      = OT1-sc,
4503 <bch> [ name      = bch-sc,
4504 <cmr> [ name      = cmr-sc-OT1,
4505 <pad> [ name      = pad-sc,
4506 <pmn> [ name      = pmnj-sc,
4507 <ppl> [ name      = ppl-sc,
4508 <ptm> [ name      = ptm-sc,
4509 <m-t>   load      = default ]
4510 <bch>   load      = bch-default ]
4511 <cmr>   load      = cmr-OT1 ]
4512 <pad>   load      = pad-default ]
4513 <pmn>   load      = pmnj-default ]
4514 <ppl>   load      = ppl-default ]
4515 <ptm>   load      = ptm-default ]
4516 <m-t|bch|pad|pmn> { encoding = OT1,
4517 <cmr|ppl|ptm> { encoding = {OT1,OT4},
4518 <bch>   family    = bch,
4519 <cmr>   family    = cmr,
4520 <pad>   family    = {pad,padx,padj},

```

```

4521 <pmn>    family    = pmnj,
4522 <ppl>    family    = {ppl,pplx,pplj},
4523 <ptm>    family    = {ptm,ptmx,ptmj},
4524    shape    = sc }
4525    {
4526        a = {50,50},
4527 <cmr|pad|ppl|ptm>    \ae = {50, },
4528 <bch|pmn>    c = {50, },
4529 <bch|pad|pmn>    d = { ,50},
4530 <m-t|bch|cmr|pad|pmn|ptm>    f = { ,50},
4531 <bch|pad|pmn>    g = {50, },
4532 <m-t|cmr|pad|pmn|ppl|ptm>    j = {50, },
4533 <bch>    j = {100, },
4534 <m-t|bch|cmr|pad|pmn|ppl>    l = { ,50},
4535 <ptm>    l = { ,80},
4536 <m-t|bch|cmr|pad|pmn|ppl>    013 = { ,50}, % fl
4537 <ptm>    013 = { ,80}, % fl
4538 <bch|pad|pmn>    o = {50,50},
4539 <pad|pmn>    \oe = {50, },
4540 <ppl>    p = { 0, 0},
4541 <bch|pad|pmn>    q = {50,70},
4542 <ppl>    q = { 0, },
4543 <m-t|cmr|pad|pmn|ppl|ptm>    r = { , 0},
4544    t = {50,50},
4545 <m-t|bch|cmr|pad|pmn|ppl>    y = {50,50},
4546 <ptm>    y = {80,80},
4547    }
4548
4549 \SetProtrusion
4550 <m-t>    [ name      = T1-sc,
4551 <bch>    [ name      = bch-sc-T1,
4552 <cmr>    [ name      = cmr-sc-T1,
4553 <pad>    [ name      = pad-sc-T1,
4554 <pmn>    [ name      = pmnj-sc-T1,
4555 <ppl>    [ name      = ppl-sc-T1,
4556 <ptm>    [ name      = ptm-sc-T1,
4557 <m-t>    load      = T1-default ]
4558 <bch>    load      = bch-T1    ]
4559 <cmr>    load      = cmr-T1    ]
4560 <pad>    load      = pad-T1    ]
4561 <pmn>    load      = pmnj-T1   ]
4562 <ppl>    load      = ppl-T1    ]
4563 <ptm>    load      = ptm-T1    ]
4564    { encoding = {T1,LY1},
4565 <bch>    family    = bch,
4566 <cmr>    family    = cmr,
4567 <pad>    family    = {pad,padx,padj},
4568 <pmn>    family    = pmnj,
4569 <ppl>    family    = {ppl,pplx,pplj},
4570 <ptm>    family    = {ptm,ptmx,ptmj},
4571    shape    = sc }
4572    {
4573        a = {50,50},
4574 <cmr|pad|ppl|ptm>    \ae = {50, },
4575 <bch|pmn>    c = {50, },
4576 <bch|pad|pmn>    d = { ,50},
4577 <m-t|bch|cmr|pad|pmn|ptm>    f = { ,50},
4578 <bch|pad|pmn>    g = {50, },
4579 <m-t|cmr|pad|pmn|ppl|ptm>    j = {50, },
4580 <bch>    j = {100, },
4581 <m-t|bch|cmr|pad|pmn|ppl>    l = { ,50},
4582 <ptm>    l = { ,80},
4583 <m-t|bch|cmr|pad|pmn|ppl>    029 = { ,50}, % fl

```

```

4584 <ptm> 029 = { ,80}, % fl
4585 <bch|pad|pmn> o = {50,50},
4586 <bch|pad|pmn> \oe = {50, },
4587 <ppl> p = { 0, 0},
4588 <bch|pad|pmn> q = {50,70},
4589 <ppl> q = { 0, },
4590 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
4591 t = {50,50},
4592 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50},
4593 <ptm> y = {80,80},
4594 }
4595
4596 </!ugm>
4597 <*m-t>
4598 \SetProtrusion
4599 [ name = QX-sc,
4600 load = QX-default ]
4601 { encoding = QX,
4602 shape = sc }
4603 {
4604 a = {50,50},
4605 f = { ,50},
4606 j = {50, },
4607 l = { ,50},
4608 013 = { ,50}, % fl
4609 r = { , 0},
4610 t = {50,50},
4611 y = {50,50},
4612 }
4613
4614 </m-t>
4615 <*cmr|bch>
4616 \SetProtrusion
4617 <bch> [ name = bch-sc-T5,
4618 <bch> load = bch-T5 ]
4619 <cmr> [ name = cmr-sc-T5,
4620 <cmr> load = cmr-T5 ]
4621 { encoding = T5,
4622 <bch> family = bch,
4623 <cmr> family = cmr,
4624 shape = sc }
4625 {
4626 a = {50,50},
4627 <bch> c = {50, },
4628 <bch> d = { ,50},
4629 f = { ,50},
4630 <bch> g = {50, },
4631 <bch> j = {100, },
4632 <cmr> j = {50, },
4633 l = { ,50},
4634 <bch> o = {50,50},
4635 <bch> q = { 0, },
4636 <cmr> r = { , 0},
4637 t = {50,50},
4638 y = {50,50},
4639 }
4640
4641 </cmr|bch>
4642 <*pmn>
4643 \SetProtrusion
4644 [ name = pmnx-sc,
4645 load = pmnj-sc ]
4646 { encoding = OT1,

```

```

4647     family    = pmnx,
4648     shape      = sc }
4649 {
4650     1 = {230,180},
4651 }
4652
4653 \SetProtrusion
4654 [ name      = pmnx-sc-T1,
4655   load      = pmnj-sc-T1 ]
4656 { encoding = {T1,LV1},
4657   family    = pmnx,
4658   shape     = sc }
4659 {
4660     1 = {230,180},
4661 }
4662

```

#### 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`.

```

4663 \SetProtrusion
4664 [ name      = pmnj-scit,
4665   load      = pmnj-it  ]
4666 { encoding = OT1,
4667   family    = pmnj,
4668   shape     = {scit,si} }
4669 {
4670     a = {50, },
4671     \ae = { , -50},
4672     b = {20, -50},
4673     c = {50, -50},
4674     d = {20, 0},
4675     e = {20, -50},
4676     f = {10, 0},
4677     012 = {10, -50}, % fi
4678     013 = {10, -50}, % fl
4679     014 = {10, -50}, % ffi
4680     015 = {10, -50}, % ffl
4681     g = {50, -50},
4682     i = {20, -50},
4683     j = {20, 0},
4684     k = {20, },
4685     l = {20, 50},
4686     m = { , -30},
4687     n = { , -30},
4688     o = {50, },
4689     \oe = {50, -50},
4690     p = {20, -50},
4691     q = {50, },
4692     r = {20, 0},
4693     s = {20, -30},
4694     t = {70, },
4695     u = {50, -50},
4696     v = {100, },
4697     w = {100, },
4698     y = {50, },
4699     z = { , -50},
4700 }
4701
4702 \SetProtrusion

```

```

4703 [ name      = pmnj-scit-T1,
4704     load      = pmnj-it-T1 ]
4705 { encoding = {T1,LV1},
4706   family   = pmnj,
4707   shape     = {scit,si} }
4708 {
4709   a = {50, },
4710   \ae = { , -50},
4711   b = {20,-50},
4712   c = {50,-50},
4713   d = {20, 0},
4714   e = {20,-50},
4715   f = {10, 0},
4716   028 = {10,-50}, % fi
4717   029 = {10,-50}, % fl
4718   030 = {10,-50}, % ffi
4719   031 = {10,-50}, % ffl
4720   g = {50,-50},
4721   i = {20,-50},
4722   188 = {20, 0}, % ij
4723   j = {20, 0},
4724   k = {20, },
4725   l = {20,50},
4726   m = { , -30},
4727   n = { , -30},
4728   o = {50, },
4729   \oe = {50,-50},
4730   p = {20,-50},
4731   q = {50, },
4732   r = {20, 0},
4733   s = {20,-30},
4734   t = {70, },
4735   u = {50,-50},
4736   v = {100, },
4737   w = {100, },
4738   y = {50, },
4739   z = { , -50},
4740 }
4741
4742 \SetProtrusion
4743 [ name      = pmnx-scit,
4744     load      = pmnj-scit ]
4745 { encoding = OT1,
4746   family   = pmnx,
4747   shape     = {scit,si} }
4748 {
4749   1 = {100,150},
4750 }
4751
4752 \SetProtrusion
4753 [ name      = pmnx-scit-T1,
4754     load      = pmnj-scit-T1 ]
4755 { encoding = {T1,LV1},
4756   family   = pmnx,
4757   shape     = {scit,si} }
4758 {
4759   1 = {100,150},
4760 }
4761
4762 </pmn>

```



### 14.7.5 textcomp

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

4763 \SetProtrusion
4764 <m-t> [ name = textcomp ]
4765 <bch> [ name = bch-textcomp ]
4766 <cmr> [ name = cmr-textcomp ]
4767 <pad> [ name = pad-textcomp ]
4768 <pmn> [ name = pmn-textcomp ]
4769 <ppl> [ name = ppl-textcomp ]
4770 <ptm> [ name = ptm-textcomp ]
4771 <ugm> [ name = ugm-textcomp ]
4772 <m-t> { encoding = TS1 }
4773 <!m-t> { encoding = TS1,
4774 <bch> family = bch }
4775 <cmr> family = cmr }
4776 <pad> family = {pad,padx,padj} }
4777 <pmn> family = {pmnx,pmnj} }
4778 <ppl> family = {ppl,pplx,pplj} }
4779 <ptm> family = {ptm,ptmx,ptmj} }
4780 <ugm> family = ugm }
4781 {
4782 <cmr> \textquotestraightbase = {300,300},
4783 <pad|pmn> \textquotestraightbase = {400,400},
4784 <cmr|pmn> \textquotestraightdblbase = {300,300},
4785 <pad> \textquotestraightdblbase = {400,400},
4786 <bch|cmr|pad|pmn|ugm> \texttwelveudash = {200,200},
4787 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
4788 <ugm> \textthreequartersemdash = {200,200},
4789 <cmr|pmn> \textquotesingle = {300,400},
4790 <pad> \textquotesingle = {400,500},
4791 <ptm> \textquotesingle = {500,500},
4792 <ugm> \textquotesingle = {300,500},
4793 <bch|cmr|pmn> \textasteriskcentered = {200,300},
4794 <pad> \textasteriskcentered = {300,300},
4795 <ugm> \textasteriskcentered = {100,200},
4796 <pmn> \textfractionsolidus = {-200,-200},
4797 <cmr> \textoneoldstyle = {100,100},
4798 <pmn> \textoneoldstyle = { , 50},
4799 <cmr> \textthreeoldstyle = { , 50},
4800 <pad|pmn> \textthreeoldstyle = { 50, },
4801 <cmr> \textfouroldstyle = { 50, 50},
4802 <pad|pmn> \textfouroldstyle = { 50, },
4803 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
4804 <cmr> \textlangle = {400, },
4805 <cmr> \textrangle = { ,400},
4806 <m-t|bch|pmn|ptm> \textminus = {200,200},
4807 <cmr|pad|ppl> \textminus = {300,300},
4808 <ugm> \textminus = {250,300},
4809 <bch|pad|pmn> \textlbrackdbl = {100, },
4810 <bch|pad|pmn> \textrbrackdbl = { ,100},
4811 <pmn> \textasciigrave = {200,500},
4812 <bch|cmr|pad|pmn> \texttildelow = {200,250},
4813 <pmn> \textasciibreve = {300,400},
4814 <pmn> \textasciicaron = {300,400},
4815 <pmn> \textacutedbl = {200,300},
4816 <pmn> \textgravedbl = {150,300},
4817 <bch|pmn|ugm> \textdagger = { 80, 80},
4818 <cmr|pad> \textdagger = {100,100},
4819 <ptm> \textdagger = {150,150},
4820 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},

```

```

4821 <ptm> \textdaggerdbl = {100,100},
4822 <bch> \textbardbl = {100,100},
4823 <ugm> \textbardbl = {150,150},
4824 <bch> \textbullet = {200,200},
4825 <cmr|pad|pmn> \textbullet = { ,100},
4826 <ptm> \textbullet = {150,150},
4827 <ugm> \textbullet = { 50,100},
4828 <bch|cmr|pmn> \textcelsius = { 50, },
4829 <pad> \textcelsius = { 80, },
4830 <bch> \textflorin = { 50, 50},
4831 <pad|ugm> \textflorin = { ,100},
4832 <pmn> \textflorin = { 50,100},
4833 <ptm> \textflorin = { 50, 70},
4834 <cmr> \textcolonmonetary = { , 50},
4835 <pad|pmn> \textcolonmonetary = { 50, },
4836 <pmn> \textinterrobang = { ,100},
4837 <pmn> \textinterrobangdown = {100, },
4838 <m-t|pad|ptm> \texttrademark = {100,100},
4839 <bch> \texttrademark = {150,150},
4840 <cmr|ppl> \texttrademark = {200,200},
4841 <pmn> \texttrademark = { 50, 50},
4842 <ugm> \texttrademark = {100,150},
4843 <bch|ugm> \textcent = { 50, },
4844 <ptm> \textcent = {100,100},
4845 <bch> \textsterling = { 50, },
4846 <ugm> \textsterling = { , 50},
4847 <bch> \textbrokenbar = {200,200},
4848 <ugm> \textbrokenbar = {200,300},
4849 <pmn> \textasciidieresis = {300,400},
4850 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
4851 <pmn> \textcopyright = {100,150},
4852 <ppl> \textcopyright = {200,200},
4853 <bch|cmr|ugm> \textordfeminine = {100,200},
4854 <pad|pmn> \textordfeminine = {200,200},
4855 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
4856 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
4857 <pmn> \textregistered = { 50,150},
4858 <ppl> \textregistered = {200,200},
4859 <pmn> \textasciimacron = {150,200},
4860 <m-t|ppl|ptm> \textdegree = {300,300},
4861 <bch> \textdegree = {150,200},
4862 <cmr|pad> \textdegree = {400,400},
4863 <pmn> \textdegree = {150,400},
4864 <ugm> \textdegree = {200,200},
4865 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
4866 <ptm> \textpm = { 50, 80},
4867 <bch|ugm> \texttwosuperior = {100,200},
4868 <cmr> \texttwosuperior = { 50,100},
4869 <pad|pmn> \texttwosuperior = {200,200},
4870 <ptm> \texttwosuperior = { 50, 50},
4871 <bch|ugm> \textthreesuperior = {100,200},
4872 <cmr> \textthreesuperior = { 50,100},
4873 <pad|pmn> \textthreesuperior = {200,200},
4874 <ptm> \textthreesuperior = { 50, 50},
4875 <pmn> \textasciicute = {300,400},
4876 <bch|ugm> \textmu = { ,100},
4877 <bch|pad|pmn> \textparagraph = { ,100},
4878 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
4879 <ptm> \textperiodcentered = {300,300},
4880 <ugm> \textperiodcentered = {200,500},
4881 <bch|ugm> \textonesuperior = {200,300},
4882 <cmr|pad|pmn> \textonesuperior = {200,200},
4883 <ptm> \textonesuperior = {100,100},

```

```

4884 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
4885 <cmr> \textordmasculine = {100,200},
4886 <bch|cmr|pmn> \texteuro = {100, },
4887 <pad> \texteuro = { 50,100},
4888 <bch|ptm> \texttimes = {100,100},
4889 <cmr> \texttimes = {150,250},
4890 <pad> \texttimes = {100,150},
4891 <pmn> \texttimes = { 70,100},
4892 <ugm> \texttimes = {200,300},
4893 <bch|pad|pmn> \textdiv = {150,200},
4894 <cmr> \textdiv = {150,250},
4895 <ptm> \textdiv = { 50,100},
4896 <ugm> \textdiv = {200,300},
4897 <ptm> \textperthousand = { ,50},
4898 <ugm> \textsection = { ,100},
4899 <ugm> \textonehalf = { 50,100},
4900 <ugm> \textonequarter = { 50,100},
4901 <ugm> \textthreequarters = { 50,100},
4902 <ugm> \textsurd = { ,100},

```

Remaining slots in the source file.

```

4903 }
4904
4905 <*cmr|pad|pmn|ugm>
4906 \SetProtrusion
4907 <cmr> [ name = cmr-textcomp-it ]
4908 <pad> [ name = pad-textcomp-it ]
4909 <pmn> [ name = pmn-textcomp-it ]
4910 <ugm> [ name = ugm-textcomp-it ]
4911 { encoding = TS1,
4912 <cmr> family = cmr,
4913 <pad> family = {pad,padx,padj},
4914 <pmn> family = {pmnx,pmnj},
4915 <ugm> family = ugm,
4916 <!ugm> shape = {it,sl} }
4917 <ugm> shape = it }
4918 {
4919 <cmr> \textquotestraightbase = {300,600},
4920 <pad|pmn> \textquotestraightbase = {400,400},
4921 <cmr> \textquotestraightdblbase = {300,600},
4922 <pad> \textquotestraightdblbase = {300,400},
4923 <pmn> \textquotestraightdblbase = {300,300},
4924 \texttwelvewardash = {200,200},
4925 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
4926 <ugm> \textthreequartersemdash = {200,200},
4927 <cmr> \textquotesingle = {600,300},
4928 <pad> \textquotesingle = {800,100},
4929 <pmn> \textquotesingle = {300,200},
4930 <ugm> \textquotesingle = {500,500},
4931 <cmr> \textasteriskcentered = {300,200},
4932 <pad> \textasteriskcentered = {500,100},
4933 <pmn> \textasteriskcentered = {200,300},
4934 <ugm> \textasteriskcentered = {300,150},
4935 <pmn> \textfractionsolidus = {-200,-200},
4936 <cmr> \textoneoldstyle = {100, 50},
4937 <pad> \textoneoldstyle = {100, },
4938 <pmn> \textoneoldstyle = { 50, },
4939 <pad> \texttwooldstyle = { 50, },
4940 <pmn> \texttwooldstyle = {-50, },
4941 <cmr> \textthreeoldstyle = {100, 50},
4942 <pmn> \textthreeoldstyle = {-100, },
4943 <cmr> \textfouroldstyle = { 50, 50},
4944 <pad> \textfouroldstyle = { 50,100},

```

```

4945 <cmr> \textsevenoldstyle = { 50, 80},
4946 <pad> \textsevenoldstyle = { 50, },
4947 <pmn> \textsevenoldstyle = { 20, },
4948 <cmr> \textlangle = {400, },
4949 <cmr> \textrangle = { ,400},
4950 <cmr|pad> \textminus = {300,300},
4951 <pmn> \textminus = {200,200},
4952 <ugm> \textminus = {250,300},
4953 <pad|pmn> \textlbrackdbl = {100, },
4954 <pad|pmn> \textrbrackdbl = { ,100},
4955 <pmn> \textasciigrave = {300,300},
4956 <cmr|pad|pmn> \texttildelow = {200,250},
4957 <pmn> \textasciibreve = {300,300},
4958 <pmn> \textasciicaron = {300,300},
4959 <pmn> \textacutedbl = {200,300},
4960 <pmn> \textgravedbl = {150,300},
4961 <cmr> \textdagger = {100,100},
4962 <pad> \textdagger = {200,100},
4963 <pmn> \textdagger = { 80, 50},
4964 <ugm> \textdagger = { 80, 80},
4965 <cmr|pad> \textdaggerdbl = { 80, 80},
4966 <pmn> \textdaggerdbl = { 80, 50},
4967 <ugm> \textbardbl = {150,150},
4968 <cmr> \textbullet = {200,100},
4969 <pad> \textbullet = {300, },
4970 <pmn> \textbullet = { 30, 70},
4971 <ugm> \textbullet = { 50,100},
4972 <cmr> \textcelsius = {100, },
4973 <pad> \textcelsius = {200, },
4974 <pmn> \textcelsius = { 50,-50},
4975 <pad> \textflorin = {100, },
4976 <pmn> \textflorin = { 50,100},
4977 <ugm> \textflorin = { ,100},
4978 <cmr> \textcolonmonetary = {150, },
4979 <pad> \textcolonmonetary = {100, },
4980 <pmn> \textcolonmonetary = { 50,-50},
4981 <cmr|pad> \texttrademark = {200, },
4982 <pmn> \texttrademark = { 50,100},
4983 <ugm> \texttrademark = {150, 50},
4984 <ugm> \textcent = { 50, },
4985 <ugm> \textsterling = { , 50},
4986 <ugm> \textbrokenbar = {200,300},
4987 <pmn> \textasciidieresis = {300,200},
4988 <cmr> \textcopyright = {100, },
4989 <pad> \textcopyright = {200,100},
4990 <pmn> \textcopyright = {100,150},
4991 <ugm> \textcopyright = {300, },
4992 <cmr> \textordfeminine = {100,100},
4993 <pmn> \textordfeminine = {200,200},
4994 <ugm> \textordfeminine = {100,200},
4995 <cmr|pad> \textlnot = {300, },
4996 <pmn|ugm> \textlnot = {200, },
4997 <cmr> \textregistered = {100, },
4998 <pad> \textregistered = {200,100},
4999 <pmn> \textregistered = { 50,150},
5000 <ugm> \textregistered = {300, },
5001 <pmn> \textasciimacron = {150,200},
5002 <cmr|pad> \textdegree = {500,100},
5003 <pmn> \textdegree = {150,150},
5004 <ugm> \textdegree = {300,200},
5005 <cmr> \textpm = {150,100},
5006 <pad> \textpm = {200,150},
5007 <pmn|ugm> \textpm = {150,200},

```

```

5008 <cmr> \textonesuperior = {400, },
5009 <pad> \textonesuperior = {300,100},
5010 <pmn> \textonesuperior = {200,100},
5011 <ugm> \textonesuperior = {300,300},
5012 <cmr> \texttwosuperior = {400, },
5013 <pad> \texttwosuperior = {300, },
5014 <pmn> \texttwosuperior = {200,100},
5015 <ugm> \texttwosuperior = {300,200},
5016 <cmr> \textthreesuperior = {400, },
5017 <pad> \textthreesuperior = {300, },
5018 <pmn> \textthreesuperior = {200,100},
5019 <ugm> \textthreesuperior = {300,200},
5020 <ugm> \textmu = { ,100},
5021 <pmn> \textasciicute = {300,200},
5022 <cmr> \textparagraph = {200, },
5023 <pmn> \textparagraph = { ,100},
5024 <cmr> \textperiodcentered = {500,500},
5025 <pad|pmn|ugm> \textperiodcentered = {300,400},
5026 <cmr> \textordmasculine = {100,100},
5027 <pmn> \textordmasculine = {200,200},
5028 <ugm> \textordmasculine = {300,200},
5029 <cmr> \texteuro = {200, },
5030 <pad> \texteuro = {100, },
5031 <pmn> \texteuro = {100,-50},
5032 <cmr> \texttimes = {200,200},
5033 <pad> \texttimes = {200,100},
5034 <pmn> \texttimes = { 70,100},
5035 <ugm> \texttimes = {200,300},
5036 <cmr|pad> \textdiv = {200,200},
5037 <pmn> \textdiv = {150,200},
5038 <ugm> \textdiv = {200,300},
5039 <ugm> \textsection = { ,200},
5040 <ugm> \textonehalf = { 50,100},
5041 <ugm> \textonequarter = { 50,100},
5042 <ugm> \textthreequarters = { 50,100},
5043 <ugm> \textsurd = { ,100},
5044 }
5045
5046 </cmr|pad|pmn|ugm>

```

### 14.7.6 Computer Modern math

Now to the math symbols for Computer Modern Roman. Definitions have been extracted from fontmath.ltx. I did not spend too much time fiddling with these settings, so they can surely be improved.

The math font ‘operators’ (also used for the `\mathrm` and `\mathbf` alphabets) is OT1/cmr, which we’ve already set up above. It’s declared as:

```

\DeclareSymbolFont{operators} {OT1}{cmr} {m}{n}
\SetSymbolFont{operators}{bold}{OT1}{cmr} {bx}{n}

```

`\mathit` (OT1/cmr/m/it) is also already set up.

There are (for the moment) no settings for `\mathsf` and `\mathtt`.

Math font ‘letters’ (also used as `\mathnormal`) is declared as:

```

\DeclareSymbolFont{letters} {OML}{cmm} {m}{it}
\SetSymbolFont{letters} {bold}{OML}{cmm} {b}{it}

```

```

5047 <*cmr>
5048 \SetProtrusion

```

```

5049 [ name      = cmr-math-letters ]
5050 { encoding = OML,
5051   family   = cmm,
5052   series   = {m,b},
5053   shape    = it   }
5054 {
5055   A = {100, 50}, % \mathnormal
5056   B = { 50,   },
5057   C = { 50,   },
5058   D = { 50, 50},
5059   E = { 50,   },
5060   F = {100, 50},
5061   G = { 50, 50},
5062   H = { 50, 50},
5063   I = { 50, 50},
5064   J = {150, 50},
5065   K = { 50,100},
5066   L = { 50, 50},
5067   M = { 50,   },
5068   N = { 50,   },
5069   O = { 50,   },
5070   P = { 50,   },
5071   Q = { 50, 50},
5072   R = { 50,   },
5073   S = { 50,   },
5074   T = { 50,100},
5075   U = { 50, 50},
5076   V = {100,100},
5077   W = { 50,100},
5078   X = { 50,100},
5079   Y = {100,100},
5080   f = {100,100},
5081   h = {   ,100},
5082   i = {   , 50},
5083   j = {   , 50},
5084   k = {   , 50},
5085   r = {   , 50},
5086   v = {   , 50},
5087   w = {   , 50},
5088   x = {   , 50},
5089   "0B = { 50,100}, % \alpha
5090   "0C = { 50, 50}, % \beta
5091   "0D = {200,150}, % \gamma
5092   "0E = { 50, 50}, % \delta
5093   "0F = { 50, 50}, % \epsilon
5094   "10 = { 50,150}, % \zeta
5095   "12 = { 50,   }, % \theta
5096   "13 = {   ,100}, % \iota
5097   "14 = {   ,100}, % \kappa
5098   "15 = {100, 50}, % \lambda
5099   "16 = {   , 50}, % \mu
5100   "17 = {   , 50}, % \nu
5101   "18 = {   , 50}, % \xi
5102   "19 = { 50,100}, % \pi
5103   "1A = { 50, 50}, % \rho
5104   "1B = {   ,150}, % \sigma
5105   "1C = { 50,150}, % \tau
5106   "1D = { 50, 50}, % \upsilon
5107   "1F = { 50,100}, % \chi
5108   "20 = { 50, 50}, % \psi
5109   "21 = {   , 50}, % \omega
5110   "22 = {   , 50}, % \varepsilon
5111   "23 = {   , 50}, % \vartheta

```

```

5112 "24 = { , 50}, % \varpi
5113 "25 = {100, }, % \varrho
5114 "26 = {100,100}, % \varsigma
5115 "27 = { 50, 50}, % \varphi
5116 "28 = {100,100}, % \leftharpoonup
5117 "29 = {100,100}, % \leftharpoondown
5118 "2A = {100,100}, % \rightharpoonup
5119 "2B = {100,100}, % \rightharpoondown
5120 "2C = {300,200}, % \lhook
5121 "2D = {200,300}, % \rhook
5122 "2E = { ,100}, % \triangleright
5123 "2F = {100, }, % \triangleleft
5124 "3A = { ,500}, % ., \ldotp
5125 "3B = { ,500}, % ,
5126 "3C = {200,100}, % <
5127 "3D = {300,400}, % /
5128 "3E = {100,200}, % >
5129 "3F = {200,200}, % \star
5130 "5B = { ,100}, % \flat
5131 "5E = {200,200}, % \smile
5132 "5F = {200,200}, % \frown
5133 "7C = {100, }, % \jmath
5134 "7D = { ,100}, % \wp

```

Remaining slots in the source file.

```

5135 }
5136

```

Math font ‘symbol’s’ (also used for the `\mathcal` alphabet) is declared as:

```

\DeclareSymbolFont{symbols} {OMS}{cmsy}{m}{n}
\SetSymbolFont{symbols} {bold}{OMS}{cmsy}{b}{n}

```

```

5137 \SetProtrusion
5138 [ name = cmr-math-symbols ]
5139 { encoding = OMS,
5140   family = cmsy,
5141   series = {m,b},
5142   shape = n }
5143 {
5144   A = {150, 50}, % \mathcal
5145   C = { ,100},
5146   D = { , 50},
5147   F = { 50,150},
5148   I = { ,100},
5149   J = {100,150},
5150   K = { ,100},
5151   L = {100, },
5152   M = { 50, 50},
5153   N = { 50,100},
5154   P = { , 50},
5155   Q = { 50, },
5156   R = { , 50},
5157   T = { 50,150},
5158   V = { 50, 50},
5159   W = { , 50},
5160   X = {100,100},
5161   Y = {100, },
5162   Z = {100,150},
5163   "00 = {300,300}, % -
5164   "01 = { ,700}, % \cdotp, \cdotp
5165   "02 = {150,250}, % \times
5166   "03 = {150,250}, % *, \ast

```

```

5167 "04 = {200,300}, % \div
5168 "05 = {150,250}, % \diamond
5169 "06 = {200,200}, % \pm
5170 "07 = {200,200}, % \mp
5171 "08 = {100,100}, % \oplus
5172 "09 = {100,100}, % \ominus
5173 "0A = {100,100}, % \otimes
5174 "0B = {100,100}, % \oslash
5175 "0C = {100,100}, % \odot
5176 "0D = {100,100}, % \bigcirc
5177 "0E = {100,100}, % \circ
5178 "0F = {100,100}, % \bullet
5179 "10 = {100,100}, % \asymp
5180 "11 = {100,100}, % \equiv
5181 "12 = {200,100}, % \subseteq
5182 "13 = {100,200}, % \supseteq
5183 "14 = {200,100}, % \leq
5184 "15 = {100,200}, % \geq
5185 "16 = {200,100}, % \preceq
5186 "17 = {100,200}, % \succeq
5187 "18 = {200,200}, % \sim
5188 "19 = {150,150}, % \approx
5189 "1A = {200,100}, % \subset
5190 "1B = {100,200}, % \supset
5191 "1C = {200,100}, % \ll
5192 "1D = {100,200}, % \gg
5193 "1E = {300,100}, % \prec
5194 "1F = {100,300}, % \succ
5195 "20 = {100,200}, % \leftarrow
5196 "21 = {200,100}, % \rightarrow
5197 "22 = {100,100}, % \uparrow
5198 "23 = {100,100}, % \downarrow
5199 "24 = {100,100}, % \leftrightarrows
5200 "25 = {100,100}, % \nearrow
5201 "26 = {100,100}, % \searrow
5202 "27 = {100,100}, % \simeq
5203 "28 = {100,100}, % \Leftarrow
5204 "29 = {100,100}, % \Rightarrow
5205 "2A = {100,100}, % \Uparrow
5206 "2B = {100,100}, % \Downarrow
5207 "2C = {100,100}, % \Leftrightarrow
5208 "2D = {100,100}, % \nrightarrow
5209 "2E = {100,100}, % \swarrow
5210 "2F = { ,100}, % \propto
5211 "30 = { ,400}, % \prime
5212 "31 = {100,100}, % \infty
5213 "32 = {150,100}, % \in
5214 "33 = {100,150}, % \ni
5215 "34 = {100,100}, % \triangle, \bigtriangleup
5216 "35 = {100,100}, % \bigtriangledown
5217 "38 = { ,100}, % \forall
5218 "39 = {100, }, % \exists
5219 "3A = {200, }, % \neg
5220 "3E = {200,200}, % \top
5221 "3F = {200,200}, % \bot, \perp
5222 "5E = {100,200}, % \wedge
5223 "5F = {100,200}, % \vee
5224 "60 = { ,300}, % \vdash
5225 "61 = {300, }, % \dashv
5226 "62 = {100,100}, % \lfloor
5227 "63 = {100,100}, % \rfloor
5228 "64 = {100,100}, % \lceil
5229 "65 = {100,100}, % \rceil

```



```

5230 "66 = {150, }, % \lbrace
5231 "67 = { ,150}, % \rbrace
5232 "68 = {400, }, % \angle
5233 "69 = { ,400}, % \rangle
5234 "6C = {100,100}, % \updownarrow
5235 "6D = {100,100}, % \Updownarrow
5236 "6E = {100,300}, % \, \backslash, \setminus
5237 "72 = {100,100}, % \nabla
5238 "79 = {200,200}, % \dagger
5239 "7A = {100,100}, % \ddagger
5240 "7B = {100, }, % \mathparagraph
5241 "7C = {100,100}, % \clubsuit
5242 "7D = {100,100}, % \diamondsuit
5243 "7E = {100,100}, % \heartsuit
5244 "7F = {100,100}, % \spadesuit

```

Remaining slots in the source file.

```

5245 }
5246

```

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}
```

```

5247 </cmr>
5248 </cfg-t>

```

### 14.7.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
5249 <*cfg-u>
```

Symbol font 'a'.

```

5250 <*msa>
5251 \SetProtrusion
5252 [ name = AMS-a ]
5253 { encoding = U,
5254   family = msa }
5255 {
5256 "05 = {150,250}, % \centerdot
5257 "06 = {100,100}, % \lozenge
5258 "07 = { 50, 50}, % \blacklozenge
5259 "08 = { 50, 50}, % \circlearrowright
5260 "09 = { 50, 50}, % \circlearrowleft
5261 "0A = {100,100}, % \rightleftharpoons
5262 "0B = {100,100}, % \leftrightharpoons
5263 "0D = {-50,200}, % \Vdash
5264 "0E = {-50,200}, % \Vvdash
5265 "0F = {-70,150}, % \vDash
5266 "10 = {100,150}, % \twoheadrightarrow
5267 "11 = {100,150}, % \twoheadleftarrow
5268 "12 = { 50,100}, % \leftleftarrows
5269 "13 = { 50, 80}, % \rightrightarrows
5270 "14 = {120,120}, % \upuparrows
5271 "15 = {120,120}, % \downdownarrows
5272 "16 = {200,200}, % \upharpoonright
5273 "17 = {200,200}, % \downharpoonright
5274 "18 = {200,200}, % \upharpoonleft
5275 "19 = {200,200}, % \downharpoonleft
5276 "1A = { 80,100}, % \rightarrowtail

```

```

5277 "1B = { 80,100}, % \leftarrowtail
5278 "1C = { 50, 50}, % \leftrightharrows
5279 "1D = { 50, 50}, % \rightleftarrows
5280 "1E = {250,  }, % \Lsh
5281 "1F = {  ,250}, % \Rsh
5282 "20 = {100,100}, % \rightsquigarrow
5283 "21 = {100,100}, % \leftrightsquigarrow
5284 "22 = {100, 50}, % \looparrowleft
5285 "23 = { 50,100}, % \looparrowright
5286 "24 = { 50, 80}, % \circeq
5287 "25 = {  ,100}, % \succsim
5288 "26 = {  ,100}, % \gtrsim
5289 "27 = {  ,100}, % \gtrapprox
5290 "28 = {150, 50}, % \multimap
5291 "2B = {100,150}, % \doteqdot
5292 "2C = {100,150}, % \triangleq
5293 "2D = {100, 50}, % \precsim
5294 "2E = {100, 50}, % \lesssim
5295 "2F = { 50, 50}, % \lessapprox
5296 "30 = {100, 50}, % \eqslantless
5297 "31 = { 50, 50}, % \eqslantgtr
5298 "32 = {100, 50}, % \curlyeqprec
5299 "33 = { 50,100}, % \curlyeqsucc
5300 "34 = {100, 50}, % \preccurlyeq
5301 "36 = { 50,  }, % \leqslant
5302 "38 = {  , 50}, % \backprime
5303 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
5304 "3C = { 50,100}, % \succcurlyeq
5305 "3E = {  , 50}, % \geqslant
5306 "40 = {  , 50}, % \sqsubset
5307 "41 = { 50,  }, % \sqsupset
5308 "42 = {  ,150}, % \vartriangleright, \rhd
5309 "43 = {150,  }, % \vartriangleleft, \lhd
5310 "44 = {  ,100}, % \trianglerighteq, \unrhd
5311 "45 = {100,  }, % \trianglelefteq, \unlhd
5312 "46 = {100,100}, % \bigstar
5313 "48 = { 50, 50}, % \blacktriangledown
5314 "49 = {  ,100}, % \blacktriangleright
5315 "4A = {100,  }, % \blacktriangleleft
5316 "4B = {  ,150}, % \dashrightarrow (the arrow)
5317 "4C = {150,  }, % \dashleftarrow
5318 "4D = { 50, 50}, % \vartriangle
5319 "4E = { 50, 50}, % \blacktriangle
5320 "4F = { 50, 50}, % \triangledown
5321 "50 = { 50, 50}, % \eqcirc
5322 "56 = {  ,150}, % \Rrightarrow
5323 "57 = {150,  }, % \Lleftarrow
5324 "58 = {100,300}, % \checkmark
5325 "5C = { 50, 50}, % \angle
5326 "5D = { 50, 50}, % \measuredangle
5327 "5E = { 50, 50}, % \sphericalangle
5328 "5F = {  , 50}, % \varpropto
5329 "60 = {100,100}, % \smallsmile
5330 "61 = {100,100}, % \smallfrown
5331 "62 = { 50,  }, % \Subset
5332 "63 = {  , 50}, % \Supset
5333 "66 = {150,150}, % \curlywedge
5334 "67 = {150,150}, % \curlyvee
5335 "68 = { 50,150}, % \leftthreetimes
5336 "69 = {100, 50}, % \rightthreetimes
5337 "6C = { 50, 50}, % \bumpeq
5338 "6D = { 50, 50}, % \Bumpeq
5339 "6E = {100,  }, % \lll

```

```

5340 "6F = { ,100}, % \ggg
5341 "70 = { 50,100}, % \ulcorner
5342 "71 = {100, 50}, % \urcorner
5343 "75 = {150,200}, % \dotplus
5344 "76 = { 50,100}, % \backsim
5345 "78 = { 50,100}, % \llcorner
5346 "79 = {100, 50}, % \lrcorner
5347 "7C = {100,100}, % \intercal
5348 "7D = { 50, 50}, % \circledcirc
5349 "7E = { 50, 50}, % \circledast
5350 "7F = { 50, 50}, % \circleddash

```

Remaining slots in the source file.

```

5351 }
5352
5353 </msa>

```

Symbol font ‘b’.

```

5354 <*msb>
5355 \SetProtrusion
5356 [ name = AMS-b ]
5357 { encoding = U,
5358   family = msb }
5359 {
5360   A = { 50, 50}, % \mathbb
5361   C = { 50, 50},
5362   G = { , 50},
5363   L = { , 50},
5364   P = { , 50},
5365   R = { , 50},
5366   T = { , 50},
5367   V = { 50, 50},
5368   X = { 50, 50},
5369   Y = { 50, 50},
5370 "00 = { 50, 50}, % \lvertneqq
5371 "01 = { 50, 50}, % \gvertneqq
5372 "02 = { 50, 50}, % \nleq
5373 "03 = { 50, 50}, % \ngeq
5374 "04 = {100, 50}, % \nless
5375 "05 = { 50,150}, % \ngtr
5376 "06 = {100, 50}, % \nprec
5377 "07 = { 50,150}, % \nsucc
5378 "08 = { 50, 50}, % \lneqq
5379 "09 = { 50, 50}, % \gneqq
5380 "0A = {100,100}, % \nleqslant
5381 "0B = {100,100}, % \ngeqslant
5382 "0C = {100, 50}, % \lneq
5383 "0D = { 50,100}, % \gneq
5384 "0E = {100, 50}, % \npreceq
5385 "0F = { 50,100}, % \nsucceq
5386 "10 = { 50, }, % \precnsim
5387 "11 = { 50, 50}, % \succnsim
5388 "12 = { 50, 50}, % \lnsim
5389 "13 = { 50, 50}, % \gnsim
5390 "14 = { 50, 50}, % \leqq
5391 "15 = { 50, 50}, % \geqq
5392 "16 = { 50, 50}, % \precneqq
5393 "17 = { 50, 50}, % \succneqq
5394 "18 = { 50, 50}, % \precnapprox
5395 "19 = { 50, 50}, % \succnapprox
5396 "1A = { 50, 50}, % \lnapprox
5397 "1B = { 50, 50}, % \gnapprox
5398 "1C = {150,200}, % \nsim

```

```

5399 "1D = { 50, 50}, % \ncong
5400 "1E = {100,150}, % \diagup
5401 "1F = {100,150}, % \diagdown
5402 "20 = {100, 50}, % \varsubsetneq
5403 "21 = { 50,100}, % \varsupsetneq
5404 "22 = {100, 50}, % \nsubseteqq
5405 "23 = { 50,100}, % \nsupseteqq
5406 "24 = {100, 50}, % \subseteqqq
5407 "25 = { 50,100}, % \supseteqqq
5408 "26 = {100, 50}, % \varsubsetneqq
5409 "27 = { 50,100}, % \varsupsetneqq
5410 "28 = {100, 50}, % \subsetneq
5411 "29 = { 50,100}, % \supsetneq
5412 "2A = {100, 50}, % \nsubseteq
5413 "2B = { 50,100}, % \nsupseteq
5414 "2C = { 50,100}, % \nparallel
5415 "2D = {100,150}, % \nmid
5416 "2E = {150,150}, % \nshortmid
5417 "2F = {100,100}, % \nshortparallel
5418 "30 = { ,150}, % \nvDash
5419 "31 = { ,150}, % \nVdash
5420 "32 = { ,100}, % \nvDash
5421 "33 = { ,100}, % \nVDash
5422 "34 = { ,100}, % \ntrianglerighteq
5423 "35 = {100, }, % \trianglelefteq
5424 "36 = {100, }, % \triangleleft
5425 "37 = { ,100}, % \trianglerighteq
5426 "38 = {100,200}, % \leftarrow
5427 "39 = {100,200}, % \rightarrow
5428 "3A = {100,100}, % \Leftarrow
5429 "3B = { 50,100}, % \Rightarrow
5430 "3C = {100,100}, % \Leftrightarrow
5431 "3D = {100,200}, % \leftrightharrow
5432 "3E = { 50, 50}, % \divideontimes
5433 "3F = { 50, 50}, % \varnothing
5434 "60 = {200, }, % \Finv
5435 "61 = { , 50}, % \Game
5436 "68 = {100,100}, % \eqsim
5437 "69 = { 50, }, % \beth
5438 "6A = { 50, }, % \gimel
5439 "6B = {150, }, % \daleth
5440 "6C = {200, }, % \lessdot
5441 "6D = { ,200}, % \gtrdot
5442 "6E = {100,200}, % \ltimes
5443 "6F = {150,100}, % \rtimes
5444 "70 = { 50,100}, % \shortmid
5445 "71 = { 50, 50}, % \shortparallel
5446 "72 = {200,300}, % \smallsetminus
5447 "73 = {100,200}, % \thicksim
5448 "74 = { 50,100}, % \thickapprox
5449 "75 = { 50, 50}, % \approx
5450 "76 = { 50,100}, % \succapprox
5451 "77 = { 50, 50}, % \precapprox
5452 "78 = {100,100}, % \curvearrowleft
5453 "79 = { 50,150}, % \curvearrowright
5454 "7A = { 50,200}, % \digamma
5455 "7B = {100, 50}, % \varkappa
5456 "7F = {200, }, % \backepsilon

```

Remaining slots in the source file.

```

5457 }
5458
5459 </msb>

```

### 14.7.8 Euler

Euler Roman font (package `euler`).

```

5460 *eur
5461 \SetProtrusion
5462 [ name = euler ]
5463 { encoding = U,
5464   family = eur }
5465 {
5466   "01 = {100,100},
5467   "03 = {100,150},
5468   "06 = { ,100},
5469   "07 = {100,150},
5470   "08 = {100,100},
5471   "0A = {100,100},
5472   "0B = { , 50},
5473   "0C = { ,100},
5474   "0D = {100,100},
5475   "0E = { ,100},
5476   "0F = {100,100},
5477   "10 = {100,100},
5478   "13 = { ,100},
5479   "14 = { ,100},
5480   "15 = { , 50},
5481   "16 = { , 50},
5482   "17 = { 50,100},
5483   "18 = { 50,100},
5484   "1A = { , 50},
5485   "1B = { , 50},
5486   "1C = { 50,100},
5487   "1D = { 50,100},
5488   "1E = { 50,100},
5489   "1F = { 50,100},
5490   "20 = { , 50},
5491   "21 = { , 50},
5492   "22 = { 50,100},
5493   "24 = { , 50},
5494   "27 = { 50,100},
5495   1 = {100,100},
5496   7 = { 50,100},
5497   "3A = {300,500},
5498   "3B = {200,400},
5499   "3C = {200,100},
5500   "3D = {200,200},
5501   "3E = {100,200},
5502   A = { ,100},
5503   D = { , 50},
5504   J = { 50, },
5505   K = { , 50},
5506   L = { , 50},
5507   Q = { , 50},
5508   T = { 50, },
5509   X = { 50, 50},
5510   Y = { 50, },
5511   h = { , 50},
5512   k = { , 50},
5513 }
5514
```

Extended by the `eulervm` package.

```

5515 \SetProtrusion
5516 [ name = euler-vm,
```

```

5517     load      = euler ]
5518     { encoding = U,
5519       family   = zeur }
5520     {
5521       "28 = {100,200},
5522       "29 = {100,200},
5523       "2A = {100,150},
5524       "2B = {100,150},
5525       "2C = {200,300},
5526       "2D = {200,300},
5527       "2E = {   ,100},
5528       "2F = {100,   },
5529       "3F = {150,150},
5530       "5B = {   ,100},
5531       "5E = {100,100},
5532       "5F = {100,100},
5533       "80 = {   , 50},
5534       "81 = {200,250},
5535       "82 = {100,200},
5536     }
5537
5538 </eur>

```

Euler Script font (euca1).

```

5539 <*eus>
5540 \SetProtrusion
5541 [ name      = euscript ]
5542 { encoding = U,
5543   family   = eus }
5544 {
5545     A = {100,100},
5546     B = { 50,100},
5547     C = { 50, 50},
5548     D = { 50,100},
5549     E = { 50,100},
5550     F = { 50,   },
5551     G = { 50,   },
5552     H = {   ,100},
5553     K = {   , 50},
5554     L = {   ,150},
5555     M = {   , 50},
5556     N = {   , 50},
5557     O = { 50, 50},
5558     P = { 50, 50},
5559     T = {   ,100},
5560     U = {   , 50},
5561     V = { 50, 50},
5562     W = { 50, 50},
5563     X = { 50, 50},
5564     Y = { 50,   },
5565     Z = { 50,100},
5566     "00 = {250,250},
5567     "18 = {200,200},
5568     "3A = {200,150},
5569     "40 = {   ,100},
5570     "5E = {100,100},
5571     "5F = {100,100},
5572     "66 = { 50,   },
5573     "67 = {   , 50},
5574     "6E = {200,200},
5575   }
5576
5577 \SetProtrusion

```

```
5578 [ name      = euscript-vm,
5579     load      = euscript ]
5580 { encoding = U,
5581   family   = zeus }
5582 {
5583   "01 = {600,600},
5584   "02 = {200,200},
5585   "03 = {200,200},
5586   "04 = {200,200},
5587   "05 = {150,150},
5588   "06 = {200,200},
5589   "07 = {200,200},
5590   "08 = {100,100},
5591   "09 = {100,100},
5592   "0A = {100,100},
5593   "0B = {100,100},
5594   "0C = {100,100},
5595   "0D = {100,100},
5596   "0E = {150,150},
5597   "0F = {100,100},
5598   "10 = {150,150},
5599   "11 = {100,100},
5600   "12 = {150,100},
5601   "13 = {100,150},
5602   "14 = {150,100},
5603   "15 = {100,150},
5604   "16 = {200,100},
5605   "17 = {100,200},
5606   "19 = {150,150},
5607   "1A = {150,100},
5608   "1B = {100,150},
5609   "1C = {100,100},
5610   "1D = {100,100},
5611   "1E = {250,100},
5612   "1F = {100,250},
5613   "20 = {150,200},
5614   "21 = {150,200},
5615   "22 = {150,150},
5616   "23 = {150,150},
5617   "24 = {100,200},
5618   "25 = {150,150},
5619   "26 = {150,150},
5620   "27 = {100,100},
5621   "28 = {100,100},
5622   "29 = {100,150},
5623   "2A = {100,100},
5624   "2B = {100,100},
5625   "2C = {100,100},
5626   "2D = {150,150},
5627   "2E = {150,150},
5628   "2F = {100,100},
5629   "30 = {100,100},
5630   "31 = {100,100},
5631   "32 = {100,100},
5632   "33 = {100,100},
5633   "34 = {100,100},
5634   "35 = {100,100},
5635   "3E = {150,150},
5636   "3F = {150,150},
5637   "60 = { ,200},
5638   "61 = {200, },
5639   "62 = {100,100},
5640   "63 = {100,100},
```

```

5641 "64 = {100,100},
5642 "65 = {100,100},
5643 "68 = {300, },
5644 "69 = { ,300},
5645 "6C = {100,100},
5646 "6D = {100,100},
5647 "6F = {100,100},
5648 "72 = {100,100},
5649 "73 = {200,100},
5650 "76 = { ,100},
5651 "77 = {100, },
5652 "78 = { 50, 50},
5653 "79 = {100,100},
5654 "7A = {100,100},
5655 "7D = {150,150},
5656 "7E = {100,100},
5657 "A8 = {100,100},
5658 "A9 = {100,100},
5659 "AB = {200,200},
5660 "BA = { ,200},
5661 "BB = { ,200},
5662 "BD = {200,200},
5663 "DE = {200,200},
5664 }
5665
5666 </eus>

```

Euler Fraktur font (eufrak).

```

5667 < *euf>
5668 \SetProtrusion
5669 [ name = mathfrak ]
5670 { encoding = U,
5671 family = euf }
5672 {
5673 A = { , 50},
5674 B = { , 50},
5675 C = { 50, 50},
5676 D = { , 80},
5677 E = { 50, },
5678 G = { , 50},
5679 L = { , 80},
5680 O = { , 50},
5681 T = { , 80},
5682 X = { 80, 50},
5683 Z = { 80, 50},
5684 b = { , 50},
5685 c = { , 50},
5686 k = { , 50},
5687 p = { , 50},
5688 q = { 50, },
5689 v = { , 50},
5690 w = { , 50},
5691 x = { , 50},
5692 1 = {100,100},
5693 2 = { 80, 80},
5694 3 = { 80, 50},
5695 4 = { 80, 50},
5696 7 = { 50, 50},
5697 "12 = {500,500},
5698 "13 = {500,500},
5699 ! = { ,200},
5700 ' = {200,300},
5701 ( = {200, },

```



```

5702     ) = { ,200},
5703     * = {200,200},
5704     + = {200,250},
5705     - = {200,200},
5706     {,} = {300,300},
5707     . = {400,400},
5708     {=} = {200,200},
5709     : = { ,200},
5710     ; = { ,200},
5711     ] = { ,200},
5712   }
5713
5714 </euf>
5715 </cfg-u>

```

### 14.7.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym<sup>17</sup>).

```

5716 <cfg-e>
5717 \SetProtrusion
5718 <zpeu|euroitc> { encoding = U,
5719 <mvs> { encoding = {OT1,U},
5720 <zpeu> family = zpeu }
5721 <euroitc> family = {euroitc,euroitcs} }
5722 <mvs> family = mvs }
5723 {
5724 <zpeu> E = {50, }
5725 <euroitc> E = {100,50}
5726 <mvs> 164 = {50,50}, % \EUR
5727 <mvs> 068 = {50,-100}, % \EURdig
5728 }
5729
5730 <*zpeu|euroitc>
5731 \SetProtrusion
5732 { encoding = U,
5733 <zpeu> family = zpeu,
5734 <euroitc> family = {euroitc,euroitcs},
5735 shape = it* }
5736 {
5737 <zpeu> E = {100,-50}
5738 <euroitc> E = {100,}
5739 }
5740
5741 </zpeu|euroitc>
5742 <*zpeu>
5743 \SetProtrusion
5744 { encoding = U,
5745 family = {zpeus,eurosans} }
5746 {
5747 E = {100,50}
5748 }
5749
5750 \SetProtrusion
5751 { encoding = U,
5752 family = {zpeus,eurosans},
5753 shape = it* }
5754 {
5755 E = {200, }

```

<sup>17</sup> Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

```

5756     }
5757
5758 </zpeu>
5759 </cfg-e>

```

## 14.8 Interword Spacing

Default unit is space.

```

5760 <*beta>
5761 <*m-t>
5762 %%% -----
5763 %%% INTERWORD SPACING SETTINGS
5764
5765 \SetExtraSpacing
5766 [ name = default ]
5767 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
5768 {

```

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

```

5769           {,} = { , -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)

```

5770           r = { , -300,300},

```

- before or after lowercase characters with ascenders

```

5771           b = { , -200,200},
5772           d = { , -200,200},
5773           f = { , -200,200},
5774           h = { , -200,200},
5775           k = { , -200,200},
5776           l = { , -200,200},
5777           t = { , -200,200},

```

- before of after lowercase characters with x-height plus descender with additional optical space, e. g., ‘v’, or ‘w’

```

5778           c = { , -100,100},
5779           p = { , -100,100},
5780           v = { , -100,100},
5781           w = { , -100,100},
5782           z = { , -100,100},

```

```

5783         x = { , -100, 100 },
5784         y = { , -100, 100 }, % ?

```

- before of after lowercase characters with x-height plus descender without additional optical space

```

5785         i = { , 50, -50 },
5786         m = { , 50, -50 },
5787         n = { , 50, -50 },
5788         u = { , 50, -50 },

```

- after colon and semicolon

```

5789         : = { , 200, -200 },
5790         ; = { , 200, -200 },

```

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

```

5791         . = { , 250, -250 },
5792         ! = { , 250, -250 },
5793         ? = { , 250, -250 },

```

The order has to be reversed when enlarging is needed.’

```

5794     }
5795

```

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 `TEXbook`:

‘If the space factor  $f$  is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if  $f \geq 2000$ . [...] Then the stretch component is multiplied by  $f / 1000$ , while the shrink component is multiplied by  $1000 / f$ .’

The ‘extra space’ (`\fontdimen 7`) for Computer Modern Roman is a third of `\fontdimen 2`, i. e., 333.

```

5796 \SetExtraSpacing
5797 [ name = nonfrench-cmr,
5798   load = default,
5799   context = nonfrench ]
5800 { encoding = {OT1,T1,LY1,OT4,QX,T5},
5801   family = cmr }
5802 {

```

`latex.ltx` has:

```

\def\nonfrenchspacing{
  \sfcode\ . 3000

```

```
5803      . = {333,2000,-667},
```

```
\sfcode`~? 3000
```

```
5804      ? = {333,2000,-667},
```

```
\sfcode`~! 3000
```

```
5805      ! = {333,2000,-667},
```

```
\sfcode`~: 2000
```

```
5806      : = {333,1000,-500},
```

```
\sfcode`~; 1500
```

```
5807      ; = {      , 500,-333},
```

```
\sfcode`~, 1250
```

```
5808      {,}= {      , 250,-200},
```

```
}
```

```
5809      }
```

```
5810
```

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.

```
5811 \SetExtraSpacing
5812   [ name      = nonfrench-default,
5813     load       = default,
5814     context    = nonfrench ]
5815   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
5816   {
5817     . = {240,2000,-667},
5818     ? = {240,2000,-667},
5819     ! = {240,2000,-667},
5820     : = {240,1000,-500},
5821     ; = {      , 500,-333},
5822     {,}= {      , 250,-200},
5823   }
5824
```

## 14.9 Additional Kerning

Default unit is 1em.

```
5825 %%% -----
5826 %%% ADDITIONAL KERNING
5827
```

A dummy list to be loaded when no context is active.

```
5828 \SetExtraKerning
5829   [ name = empty ]
5830   { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
5831   { }
5832
```

### 14.9.1 French

For the French context.

```
5833 \SetExtraKerning
5834 [ name      = french-default,
5835   context   = french,
5836   unit      = space ]
5837 { encoding = {OT1,T1,LY1} }
5838 {
5839   : = {1000,}, % = \fontdimen2
5840   ; = {500, }, % ~ \thinspace
5841   ! = {500, },
5842   ? = {500, },
5843 }
5844
```

This has the disadvantage that the word following a left guillemet will not be hyphenated. This might be fixed in pdf $\TeX$ .

```
5845 \SetExtraKerning
5846 [ name      = french-guillemets,
5847   context   = french-guillemets,
5848   load      = french-default,
5849   unit      = space ]
5850 { encoding = {T1,LY1} }
5851 {
5852   \guillemotleft = { ,800}, % = 0.8\fontdimen2
5853   \guillemotright = {800, },
5854 }
5855
5856 \SetExtraKerning
5857 [ name      = french-guillemets-OT1,
5858   context   = french-guillemets,
5859   load      = french-default,
5860   unit      = space ]
5861 { encoding = OT1 }
5862 { }
5863
```

### 14.9.2 Turkish

And for Turkish.

```
5864 \SetExtraKerning
5865 [ name      = turkish,
5866   context   = turkish,
5867   unit      = space ]
5868 { encoding = {OT1,T1,LY1} }
5869 {
5870   : = {500, }, % ~ \thinspace
5871   ! = {500, },
5872   {=} = {500, },
5873 }
5874
```

### 14.9.3 Letterspacing

The settings with the ‘letterspacing’ context will be loaded whenever the command `\textls` resp. `\lsstyle` are used.

```
5875 %%% The following settings with `context=letterspacing'
5876 %%% will be loaded by \lsstyle and \textls:
```

5877

No additional kerning will be applied in math mode, but we don't want confusing error messages (e. g., when writing L<sup>A</sup>T<sub>E</sub>X).

```
5878 \SetExtraKerning
5879   [ name      = letterspacing-all,
5880     context   = letterspacing,
5881     preset    = {1000,1000} ]
5882   { encoding = {U,OML,OMS,OMX} }
5883   { }
5884
```

The full stop should be spaced out less. Numbers are not spaced out, according to soul.

```
5885 \SetExtraKerning
5886   [ name      = letterspacing-text,
5887     context   = letterspacing,
5888     preset    = {1000,1000} ]
5889   { }
5890   {
5891     . = {0, },
5892     0 = {0,0},
5893     1 = {0,0},
5894     2 = {0,0},
5895     3 = {0,0},
5896     4 = {0,0},
5897     5 = {0,0},
5898     6 = {0,0},
5899     7 = {0,0},
5900     8 = {0,0},
5901     9 = {0,0},
5902   }
5903
```

Also, quotation marks receive only half the kerning.

```
5904 \SetExtraKerning
5905   [ name      = letterspacing-OT1,
5906     context   = letterspacing,
5907     load      = letterspacing-text,
5908     preset    = {1000,1000} ]
5909   { encoding = {OT1,OT4,QX} }
5910   {
5911     \textquoteleft = {0,0}, \textquoteright = {0,0},
5912     \textquotedblleft = {0,0}, \textquotedblright = {0,0},
5913   }
5914
5915 \SetExtraKerning
5916   [ name      = letterspacing-T1,
5917     context   = letterspacing,
5918     load      = letterspacing-text,
5919     preset    = {1000,1000} ]
5920   { encoding = {T1,LV1,T5} }
5921   {
5922     \textquoteleft = {0,0}, \textquoteright = {0,0},
5923     \textquotedblleft = {0,0}, \textquotedblright = {0,0},
5924     \quotesinglbase = {0,0}, \quotedblbase = {0,0},
5925     \guilsinglleft = {0,0}, \guilsinglright = {0,0},
5926     \guillemotleft = {0,0}, \guillemotright = {0,0},
5927   }
5928
5929 /m-t
5930 /beta
```

5931 *</config>*

## 15 Auxiliary File for Micro Fine Tuning

This file can be used to test protrusion and expansion settings.

```

5932 <*test>
5933 \documentclass{article}
5934
5935 %% Here you can specify the font you want to test, using
5936 %% the commands \fontfamily, \fontseries and \fontshape.
5937 %% Make sure to end all lines with a comment character!
5938 \newcommand*{\TestFont{%
5939   \fontfamily{ppl}%
5940   \fontseries{b}%
5941   \fontshape{it}% sc, sl
5942 }
5943
5944 \usepackage{ifthen}
5945 \usepackage[T1]{fontenc}
5946 \usepackage[latin1]{inputenc}
5947 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
5948
5949 \pagestyle{empty}
5950 \setlength{\parindent}{0pt}
5951 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
5952 \newcommand*\testprotrusion[2][{}]{%
5953   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
5954   lorem ipsum dolor sit amet,
5955   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
5956   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
5957   you know the rest%
5958   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
5959   \linebreak
5960   {\fontencoding{\encodingdefault}%
5961    \fontseries{\seriesdefault}%
5962    \fontshape{\shapedefault}%
5963    \selectfont
5964    Here is the beginning of a line, \dotfill and here is its end}\linebreak
5965 }
5966 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
5967 \def\stripprefix#1>{}
5968 \newcount\charcount
5969 \begin{document}
5970
5971 \microtypesetup{expansion=false}
5972
5973 {\centering The font in this document is called by:\\
5974 \texttt{\showTestFont}\par}\bigskip
5975
5976 \TestFont\selectfont
5977 This line intentionally left empty\linebreak
5978 %% A -- Z
5979 \charcount=65
5980 \loop
5981   \testprotrusion{\char\charcount}
5982   \advance\charcount 1
5983   \ifnum\charcount < 91 \repeat
5984 %% a -- z
5985 \charcount=97
5986 \loop

```

```

5987 \testprotrusion{\char\charcount}
5988 \advance\charcount 1
5989 \ifnum\charcount < 123 \repeat
5990 %% 0 -- 9
5991 \charcount=48
5992 \loop
5993 \testprotrusion{\char\charcount}
5994 \advance\charcount 1
5995 \ifnum\charcount < 58 \repeat
5996 %%
5997 \testprotrusion[r]{,}
5998 \testprotrusion[r]{.}
5999 \testprotrusion[r]{;}
6000 \testprotrusion[r]{:}
6001 \testprotrusion[r]{?}
6002 \testprotrusion[r]{!}
6003 \testprotrusion[l]{\textexclamdown}
6004 \testprotrusion[l]{\textquestiondown}
6005 \testprotrusion[r]{)}
6006 \testprotrusion[l]{(}
6007 \testprotrusion{/}
6008 \testprotrusion{\char~\}
6009 \testprotrusion{-}
6010 \testprotrusion{\textendash}
6011 \testprotrusion{\textemdash}
6012 \testprotrusion{\textquoteleft}
6013 \testprotrusion{\textquoteright}
6014 \testprotrusion{\textquotedblleft}
6015 \testprotrusion{\textquotedblright}
6016 \testprotrusion{\quotesinglbase}
6017 \testprotrusion{\quotedblbase}
6018 \testprotrusion{\guilsinglleft}
6019 \testprotrusion{\guilsinglright}
6020 \testprotrusion{\guillemotleft}
6021 \testprotrusion{\guillemotright}
6022
6023 \newpage
6024 The following displays the current font stretched by 5%,
6025 normal, and shrunk by 5%:
6026
6027 \bigskip
6028 \newlength{\MTln}
6029 \newcommand*\teststring
6030 {ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
6031 \settowidth{\MTln}{\teststring}
6032 \microtypesetup{expansion=true}
6033
6034 \parbox{1.05\MTln}{\teststring\linebreak\}
6035 \parbox{0.95\MTln}{\teststring\par\bigskip}
6036 \parbox{0.95\MTln}{\teststring}
6037
6038 \end{document}
6039 /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> ) . . . . .	56	<code>\MT@get@basefamily</code> : only remove suffix, if it is 'x' or 'j' . . . . .	57
issue an error instead of a warning, when pdfTeX version is too old for autoexpand . . . . .	98	<code>\MT@get@listname@</code> : don't check for empty attributes list . . . . .	58
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i> ) . . . . .	103	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i> ) . . . . .	30
Protrusion: add factors for some more characters settings for Adobe Minion (contributed by <i>Harald Harders</i> ) . . . . .	109	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i> ) . . . . .	61
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance . . . . .	82	<code>\MT@pdfTeX@no</code> : fix concerning version check (reported by <i>Harald Harders</i> ) . . . . .	26
<code>\MT@declare@sets</code> : remove spaces around set name . . . . .	71	<code>\MT@permute</code> : don't use sets for empty encoding . . . . .	84
<code>\MT@DeclareSet@</code> : remove spaces around first argument . . . . .	70	<code>\MT@pr@split</code> : fix: allow zero and negative values . . . . .	43
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded . . . . .	56	<code>\MT@use@set</code> : remove spaces around set name . . . . .	75
		<code>\UseMicrotypeSet</code> : remove spaces around first argument . . . . .	75

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

General: check for packages that might load fonts . . . . .	66	<code>\MT@get@highlevel</code> : check whether defaults have changed . . . . .	72
check whether only one encoding specified . . . . .	83	<code>\MT@get@listname@</code> : alternatively check for alias font name . . . . .	58
Font Sets: declare cmr as an alias of cmr . . . . .	102	<code>\MT@get@size</code> : additional magic to catch some errors hijack <code>\set@fontsize</code> instead of <code>\@setfontsize</code> . . . . .	73
new: allmath and basicmath . . . . .	101	<code>\MT@get@slot</code> : fix: group must also include <code>\MT@get@composite</code> . . . . .	60
Protrusion: add settings for Adobe Garamond and Computer Modern Roman in TS1 encoding . . . . .	129	<code>\MT@loop</code> : fix: new macro, used instead of <code>\loop</code> . . . . .	34
add settings for Computer Modern Roman math symbols . . . . .	133	<code>\MT@maybe@do</code> : also check for alias font name . . . . .	39
<code>\MT@context</code> : fix: set inheritance list globally to <code>\@empty</code> . . . . .	60	<code>\MT@permute@@@@</code> : more sanity checks for <code>\SetProtrusion</code> and <code>\SetExpansion</code> . . . . .	85
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement . . . . .	39	<code>\MT@setupfont</code> : also search for alias font file . . . . .	39
<code>\MT@get@basefamily</code> : also remove 'w' (swash capitals) . . . . .	57	fix: call <code>\@enc@update</code> if necessary . . . . .	39

### Version 1.3 (2004/10/27)

Font Sets: declare aer, zer and hfor as an alias of cmr . . . . .	102	<code>\MT@define@opt@keys</code> : fix: specifying load option does no longer require to give a name, too . . . . .	80
<code>\MT@catcodes</code> : check some category codes (compatibility with german) . . . . .	27	<code>\MT@load@list</code> : check whether list exists . . . . .	56

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

General: don't use scratch registers in global definitions . . . . .	60	(OT1, T1, lmr) . . . . .	113
no need to check for packages that might load fonts anymore . . . . .	66	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options . . . . .	93
use <code>\pickup@font</code> instead of <code>\define@newfont</code> as the hook for <code>\MT@setupfont</code> . . . . .	66	<code>\MT@pdfcprot@error</code> : check for pdfcprot . . . . .	36
use one instead of five counters . . . . .	34	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too . . . . .	78
Protrusion: tweak quote characters for cmr variants			

**Version 1.4a (2004/11/17)**

General: new option: <code>final</code> . . . . .	90	when reading files (reported by <i>Michael Hoppe</i> ) . . . . .	57
<code>\MT@begin@catcodes</code> : fix: reset some more catcodes			

**Version 1.4b (2004/11/26)**

General: fix: set catcodes before reading global configuration file (reported by <i>Christoph Bier</i> ) . . . . .	92	name if encoding failed . . . . .	41
new message if <code>\pdfoutput</code> is changed . . . . .	96	<code>\MT@get@basefamily</code> : fix: failed for font names of the form <code>abczz</code> (reported by <i>Georg Verwey</i> ) . . . . .	57
optimization: use <code>less</code> <code>\csnames</code> and <code>\expandafters</code> . . . . .	29	<code>\MT@get@slot</code> : don't define <code>\MT@char</code> globally (save stack problem) . . . . .	60
Protrusion: harmonize dashes in upshape and italic ( <code>cmr</code> , <code>pad</code> , <code>pp1</code> ) . . . . .	109	<code>\MT@ifdimen</code> : don't set <code>\MT@count</code> globally (save stack problem) . . . . .	31
slanted like italics . . . . .	116	<code>\MT@use@set</code> : don't use undeclared font sets . . . . .	75
<code>\MT@checklist@family</code> : fix: don't try alias family			

**Version 1.5 (2004/12/15)**

General: defaults: step: 4 (suggested by <i>Hàn Thế Thành</i> ) . . . . .	90	<code>\MT@begin@catcodes</code> : reset catcode of <code>'=</code> (compatibility with Turkish <code>babel</code> ) . . . . .	57
defaults: calculate step as <code>min(stretch,shrink)/5</code> . . . . .	97	<code>\MT@catcodes</code> : reset catcode of <code>'^</code> (compatibility with <code>chemsym</code> ) . . . . .	27
defaults: turn off expansion for DVI output . . . . .	96	<code>\MT@get@highlevel</code> : don't test defaults if called after begin document . . . . .	72
disable automatic expansion for DVI output . . . . .	97	<code>\MT@scale@factor</code> : warning for factors outside limits . . . . .	45
new option: selected, by default false (suggested by <i>Hàn Thế Thành</i> ) . . . . .	89	<code>\MT@scale@to@em</code> : don't use <code>\lcode</code> and <code>\rcode</code> for the calculation . . . . .	43
Documentation: add note about DVI output option . . . . .	7	<code>\MT@set@ex@codes</code> : allow non-selected font expansion . . . . .	48
add short history (section 12) . . . . .	21	<code>\MT@set@pr@codes</code> : adjust protrusion factors before setting the inheriting characters . . . . .	42
Inheritance: remove <code>\ss</code> from T1 list, add <code>\DJ</code> . . . . .	104		
Protrusion: settings for Bitstream Charter . . . . .	109		
<code>\DeclareMicrotypeAlias</code> : remove spaces around arguments . . . . .	76		

**Version 1.6 (2005/01/24)**

General: defaults: turn off expansion for old pdf $\TeX$ versions . . . . .	91	tune CMR math letters (OML encoding) . . . . .	133
disable automatic expansion for old pdf $\TeX$ versions . . . . .	98	<code>\MT@def@num@opt</code> : test whether numeric options receive a number . . . . .	90
load a font, if none is active . . . . .	38	<code>\MT@get@charwd</code> : use e- $\TeX$ 's <code>\fontcharwd</code> , if available . . . . .	44
new option: factor, by default 1000 . . . . .	90	<code>\MT@get@inh@list</code> : correct message if selected is false . . . . .	59
restructure dtx file . . . . .	101	<code>\MT@set@ex@codes</code> : introduce factor option . . . . .	48
test whether <code>\pickup@font</code> has changed . . . . .	67	<code>\MT@set@pr@codes</code> : introduce factor option . . . . .	42
use e- $\TeX$ 's <code>\ifcsname</code> and <code>\ifdefined</code> if defined . . . . .	30	<code>\MT@use@set</code> : retain current set if new set is undeclared . . . . .	75
Protrusion: add italic uppercase Greek letters . . . . .	116	<code>\MT@vinfo</code> : new macro: used instead of <code>\ifMT@verbose</code> . . . . .	25
improve settings for numbers (pointed out by <i>Peter Muthesius</i> ) . . . . .	111		

**Version 1.6a (2005/02/02)**

Documentation: add table of fonts with tailored protrusion settings . . . . .	15	reported by <i>Bernard Gaulle</i> ) . . . . .	60
<code>\MT@get@slot</code> : completely redone, hopefully more robust (compatible with <code>frenchpro</code> ; problem		<code>\MT@pdf@tex@no</code> : new macro . . . . .	26
		<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older pdf $\TeX$ versions . . . . .	49

**Version 1.7 (2005/03/23)**

General: \SetExpansion: bug fix: remove space after autoexpand .....	82	\MT@get@listname@: use \etfor ( <i>Andreas Böhmann's</i> idea) .....	58
\SetExpansion: don't allow automatic expansion for old pdf <sub>TeX</sub> versions .....	82	\MT@get@slot: remove backslash hack .....	60
allow specification of size ranges (suggested by <i>Andreas Böhmann</i> ) .....	72	test for \chardefed commands .....	61
modify \showhyphens .....	98	test whether \(\encoding)\(... is defined .....	60
new value for verbose option: errors .....	90	\MT@if@list@exists: don't define \MT@#1@c@name \globally, here and elsewhere .....	59
shorter command names .....	34	\MT@if@dimen: comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i> ) ...	31
warning when running in draft mode .....	95	\MT@increment: use e- <sub>TeX</sub> 's \numexpr if available ..	34
Documentation: add hint about compatibility ....	18	\MT@is@composite: new macro: construct command for composite character; no uncontrolled expansion .....	64
remove table of match order .....	11	\MT@scale: new macro: use e- <sub>TeX</sub> 's \numexpr if available .....	34
Protrusion: fix: remove \ from OT1, add \textbackslash to T1 encoding .....	112	\MT@set@ex@codes: two versions of this macro ....	48
\DeclareMicrotypeAlias: may also be used inside configuration files .....	76	\MT@split@name: don't define \MT@encoding &c. \globally .....	39
\LoadMicrotypeFile: new command (suggested by <i>Andreas Böhmann</i> ) .....	76	\MT@test@ast: make it simpler .....	72
\MicrotypeHook: new command for font package authors .....	92	\MT@try@order: always check for size, too (suggested by <i>Andreas Böhmann</i> ) .....	58
\microtypesetup: fix: warning also when setting to (no)compatibility .....	93	fix: also check for //(\series)/(\shape)// (reported by <i>Andreas Böhmann</i> ) .....	58
\MT@begin@catcodes: also use inside configuration commands .....	57	\MT@warn@code@too@large: new macro: type out maximum protrusion factor .....	45
reset catcode of ':' (compatibility with french* packages) .....	57	\MT@warn@err: new macro: for verbose=errors ...	25

**Version 1.8 (2005/06/23)**

General: \SetProtrusion: new key: unit .....	81	\MT@get@charwd: warning for missing (resp. zero-width) characters .....	44
if font substitution has occurred, set up the substitute font, not the selected one .....	66	\MT@get@dimen@six: new macro: test whether \fontdimen6 is defined .....	42
new option: config to load a different main configuration file .....	91	\MT@get@listname@: made recursive .....	58
new option: unit, by default character ....	91	\MT@get@slot: fix: expand active characters .....	60
Documentation: add example for factor option ..	12	test whether \(\encoding)\(... is defined made more robust .....	60
add example of how to get rid of a widow (suggested by <i>Adam Kucharczyk</i> ) .....	13	\MT@get@unit: new macro: get unit for codes ....	46
add hint about error messages .....	19	\MT@in@rlist: made recursive .....	33
Font Sets: add U encoding to allmath .....	101	\MT@is@active: new macro: translate inputenc-defined characters .....	62
declare pxx and txr as aliases of ppl resp. ptm	102	\MT@is@letter: warning for non-ASCII characters ..	61
Inheritance: remove \DJ from T1 list (it's the same as \DH) .....	104	\MT@led@kern: character protrusion with ledmac ..	36
Protrusion: add LY1 characters for Times .....	116	\MT@make@string: use \onelevel@sanitize .....	34
settings for AMS math fonts .....	137	\MT@map@clist@n: new macro: used instead of \etfor	32
verified settings for slanted Computer Modern Roman .....	122	\MT@map@tlist@n: new macro: used instead of \etfor	32
\DeclareMicrotypeAlias: warning when overriding an alias font .....	76	\MT@old@cmd: renamed commands from \..MicroType.. to \..Microtype.. .....	25
\DeclareMicrotypeSetDefault: new command: set default font set .....	75	\MT@orig@add@accent: fix: disable micro-typographic setup inside \add@accent (reported by <i>Stephan Hennig</i> ) .....	67
\MT@begin@catcodes: reset catcodes of the remaining ASCII characters .....	57	\MT@pdf@tex@no: case 5: pdf <sub>TeX</sub> 1.30 .....	26
\MT@check@rlist: made recursive .....	86	\MT@permute@@@@@: add ranges to the beginning of the lists .....	85
\MT@curr@list@name: new macro: current list type and name .....	65	\MT@pr@split: get character width once only ....	43
\MT@declare@sets: warning when redefining a set ..	71	\MT@scale: fix: remove spaces in non-e- <sub>TeX</sub> variant (reported by <i>Mark Rossi</i> ) .....	34
\MT@define@set@key@: use comma lists instead of token lists .....	71	\MT@setupfont@hook: restore \% and \# when hyperref is loaded .....	37
\MT@find@file: no longer wrap names in commands	56	restore csquotes's active characters .....	37

restore percent character if Spanish babel is loaded .....	37	\MT@xadd: simplified .....	32
\MT@use@set: fix: remove braces in first line .....	75		

### Version 1.9 (2005/10/28)

General: \DeclareMicrotypeSet: new key: font ..	73	settings for OT4 encoding (Computer Modern Roman, Palatino, Times) .....	109
\SetProtrusion and \SetExpansion: new key: font .....	79	settings for T5 encoding (Computer Modern Roman) .....	109
\SetProtrusion: value 'relative' renamed to 'character' for key unit .....	81	\DisableLigatures: new command: disable ligatures (requires pdfTeX 1.30) .....	75
allow context-specific font setup .....	66	\microtypecontext: new command: change setup context in the document .....	69
disable expansion if both step and shrink are zero .....	98	\MT@checklist@family: fix: add two missing \expandafters .....	41
disable microtype setup inside hyperref's \pdfstringdef (reported by <i>Hàn Thế Thành</i> ) ..	38	\MT@define@option: fix: use true as the default value ..	88
option unit: rename value relative to character .....	91	\MT@detokenize@c: fix the non-e-TeX version .....	30
warning if user requested zero step .....	97	\MT@exp@two@n: new macros: less \expandafters ..	29
Documentation: add hint about verbatim environment .....	18	\MT@get@opt: new key 'preset' to set all characters to the specified value before loading the lists ..	46
add remark about Type 1 fonts required for automatic font expansion .....	6	\MT@is@active: redone: use \set@display@protect ..	62
Font Sets: add OT4 encoding to text sets .....	101	\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning .....	61
add T5 encoding to text sets .....	101	\MT@maybe@do: redone .....	39
declare qpl and qtm (qfonts) as aliases of ppl resp. ptm .....	102	\MT@pdf@tex@no: compatibility with TeXLive hack (reported by <i>Herbert Voß</i> ) .....	26
Inheritance: add list for OT4 .....	105	\MT@rem@from@clist: new macro: remove an item from a comma list .....	33
add list for T5 (requested by <i>Hàn Thế Thành</i> ) ..	106	\MT@scale@factor: generalized .....	45
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR .....	112	\MT@toks: use instead of \toks@ .....	28

### Version 1.9a (2005/12/05)

General: new option: defersetup, by default true ..	89	diately (requested by <i>Georg Verwey</i> ) .....	71
remove superfluous test whether \pickup@font has changed .....	67	\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 .....	103	\MT@ifstreq: use \pdfstrcmp if available .....	31
Protrusion: fix: remove _ from OT1 encoding ..	113	\MT@in@clist: fix .....	33
settings for T5 encoded Charter .....	109	\MT@info@missing@char: info instead of warning (after <i>Michael Hoppe</i> reported that the 'fl' ligature is missing in Palatino SC) .....	44
\microtypesetup: inside the preamble, accepts all package options .....	93	\MT@is@feature: new macro: check for pdfTeX feature .....	35
\MT@check@font@cx: optimize context-sensitive setup ..	68	\MT@map@clist@n: following L <sup>A</sup> T <sub>E</sub> X3 .....	32
\MT@define@opt@keys: '<file name>/<line number>' as default list name .....	80	\MT@permute@@@@: don't define permutations for unused encodings .....	85
\MT@define@set@key@: don't expand variables immediately .....		\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> ) .....	38	Documentation: activate expansion in the distributed PDF .....	1
compatibility with the extendedchar option of the listings package .....	38	add samples of micro-typographic features .....	4
register with the soul package .....	38	\MT@features: use throughout the package to adjust to beta-ness .....	35

<code>\MT@ifdimen</code> : use <code>\pdfmatch</code> if available . . . . .	31	<code>present factor</code> . . . . .	45
<code>\MT@warn@code@too@large</code> : fix calculation with			

### Version 1.9c (2006/02/02)

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verweyen</i> ) . . . . .	16	<code>\MT@define@code@key@font</code> : fix: context was ignored	79
Protrusion: settings for URW Garamond . . . . .	110	<code>\MT@define@code@key@size</code> : fix: embrace <code>\MT@tempsize</code> in <code>\csname</code> (bug introduced in 1.9b) . . . . .	79

### Version 1.9d (2006/05/05)

Font Sets: <code>md*</code> instead of <code>m</code> series in basic sets . . . .	101	<code>\MT@get@font@dimen</code> : warning for zero <code>fontdimen</code> .	44
add QX encoding to text sets . . . . .	101	<code>\MT@get@opt</code> : optimize: don't reset when preset option is set . . . . .	46
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i> ) . . . . .	106	set list name before presetting . . . . .	46
Protrusion: settings for QX encoding (contributed by <i>Maciej Eder</i> ) . . . . .	114	<code>\MT@is@active</code> : support for Unicode ( <code>inputenc/utf8</code> ) . . . . .	63
settings for Euro symbols (Adobe, ITC, <i>marvosym</i> )	145	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>tex4ht</code> is loaded (reported by <i>Peter Dyballa</i> ) . . . . .	37
tweak AMS settings . . . . .	137	<code>\SetProtrusion</code> : (et al.) optimize: unify keys for mandatory argument . . . . .	77
<code>\DeclareCharacterInheritance</code> : fix: empty context	82	(et al.) split keys of optional and mandatory argument . . . . .	77
<code>\lststyle</code> : fix: font was always added to list . . . . .	55		
<code>\MT@detokenize@n</code> : new macro: use <code>\detokenize</code> if available . . . . .	30		
<code>\MT@get@ex@opt</code> : fix: evaluate preset . . . . .	50		

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

General: fix: default value for <code>activate</code> : <code>true</code> . . . .	88	fix: forgotten comma in <code>mt-mvs.cfg</code> ; adapt to <i>marvosym</i> 's changed encoding . . . . .	145
no tracing with <code>trace</code> package . . . . .	67	settings for Euler Roman font . . . . .	141
Documentation: add hint about unknown encodings include LPPL . . . . .	164	<code>\DeclareCharacterInheritance</code> : new key 'inputenc' to set the input encoding . . . . .	82
Font Sets: declare <code>zeur</code> and <code>zeus</code> ( <code>eulervm</code> ) as aliases of <code>eur</code> resp. <code>eus</code> ( <code>euler</code> ) . . . . .	103	<code>\MT@rem@from@clist</code> : model after <code>\@removeelement</code> . . . . .	33
Inheritance: adapt to <i>marvosym</i> 's changed encoding	107	<code>\MT@setup@</code> : compatibility with the <code>combine</code> class . . . . .	36
Protrusion: complete settings for Euler Fraktur and Script fonts . . . . .	144	<code>\SetExpansion</code> : new key: <code>inputenc</code> . . . . .	78
		<code>\SetProtrusion</code> : new key: <code>inputenc</code> . . . . .	77

### Version 1.9f (2006/09/09)

Protrusion: fix: <code>euler-vm</code> did not load <code>euler</code> settings	141	<code>\MT@set@inh@list</code> : fix: forgotten comma in the features list . . . . .	83
<code>\MT@curr@list@name</code> : fix: <code>\MessageBreak</code> must not be expanded . . . . .	65	<code>\MT@set@inputenc</code> : fix: input encoding must be set after the inheritance list has been parsed . . . . .	47
<code>\MT@gdef@n</code> : new macros: global variants . . . . .	29	<code>\MT@set@named@keys</code> : new macro: set name first, simplify parsing of optional argument . . . . .	79
<code>\MT@glet</code> : new macro . . . . .	29	<code>\SetProtrusion</code> : (et al.) set catcodes before parsing optional argument . . . . .	77
<code>\MT@reset@context</code> : only reset context if it has actually been changed . . . . .	69		

### Version 2.0 (N/A)

General: new option: <code>babel</code> , by default <code>false</code> (language-dependent setup suggested by <i>Ulrich Durr</i> ) . . . . .	89	<code>\MT@pdfTeX@no</code> : case 6: <code>pdfTeX 1.40</code> . . . . .	26
new option: <code>letterspacing</code> , by default <code>100</code> . . . .	90	<code>\SetExtraKerning</code> : new command: additional kerning . . . . .	78
<code>\DeclareMicrotypeBabelHook</code> : new command: interaction with <code>babel</code> . . . . .	77	<code>\SetExtraSpacing</code> : new command: adjustment of interword spacing . . . . .	78
		<code>\textls</code> : new command: <code>letterspacing</code> . . . . .	55

## 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	7
activate	5
auto	6
config	8
draft	8
expansion	5
factor	6
final	8
protrusion	5
selected	7
shrink	7
step	7
stretch	7
unit	6
verbose	8

### User Commands:

\DeclareCharacterInheritance	14
\DeclareMicrotypeAlias	15
\DeclareMicrotypeSet	8
\DeclareMicrotypeSet*	8
\DeclareMicrotypeSetDefault	10
\DisableLigatures	17
\LoadMicrotypeFile	16
\microtypecontext	16
\microtypesetup	8
\SetExpansion	13
\SetProtrusion	11
\UseMicrotypeSet	10

\@enc@update	593
--------------	-----

### A

a0poster (package)	73
activate (option)	5, 88, 157
\add@accent	1764, 1765
ae (package)	16, 102
amssymb (package)	137
article (package)	17
auto (option)	6, 89

### B

babel (option)	89, 157
babel (package)	2, 3, 37, 57, 77, 100, 101, 103, 154, 156, 157

### C

\cf@encoding	593
chemsym (package)	154
CJK (package)	19, 67
cm-super (package)	6
color (package)	7, 96
combine (package)	36, 157
config (option)	8, 15, 22, 92, 155
contour (package)	96
crop (package)	96

csquotes (package)	20, 22, 37, 112, 155
\curr@fontshape	1330, 1335, 1734

### D

\DeclareCharacterInheritance	9, 14, 2427, 3045
\DeclareMicrotypeAlias	5, 15, 31, 2152, 3046
\DeclareMicrotypeBabelHook	16, 25, 2185
\DeclareMicrotypeSet	2, 8, 20, 21, 32, 1871, 3037
\DeclareMicrotypeSet*	8, 1871
\DeclareMicrotypeSetDefault	4, 10, 23, 2124
deferssetup (option)	37, 89, 93, 156
\define@newfont	1736
\DisableLigatures	8, 17, 24, 2109
\do@subst@correction	1733
docstrip (package)	24
draft (option)	8, 89, 90
dsfont (package)	42
DVIoutput (option)	7, 21, 89, 154

### E

eco (package)	16, 102
\efcode	1029, 1063, 1064, 1108, 1111
eucal (package)	142
eufrak (package)	144
euler (package)	65, 141, 157
eulervm (package)	16, 103, 141, 157
euroitc (package)	145
europs (package)	145
eurosans (package)	145
expansion (option)	5, 6, 10, 88, 96

### F

\f@encoding	593
\f@size	1330, 1335, 1734
factor (option)	6, 12, 22, 90, 91, 154
fancyvrb (package)	18
final (option)	8, 23, 89, 90, 154
fix-cm (package)	13
\font	584
\font@name	822, 1736, 1749, 1751, 1772, 1792
\fontcharwd	801
\fontdimen	743, 750, 821, 826, 931, 934
fontinst (package)	148
fontinstallationguide (package)	127
french (package)	155
frenchpro (package)	63, 154

### G

german (package)	27, 153
graphics (package)	7, 96

### H

hfoldsty (package)	16, 102
hyperref (package)	7, 37, 38, 66, 96, 155, 156

### I

IEEEtran (package)	66
--------------------	----



- \iffontchar ..... 831  
 \ifMT@auto ..... 124, 3111, 3156  
 \ifMT@babel ..... 124, 3238, 3270  
 \ifMT@do ..... 612, 695, 1376  
 \ifMT@document ..... 165, 2546  
 \ifMT@draft ..... 124, 3025  
 \ifMT@expansion ..... 124, 3075, 3153  
 \ifMT@if@ ..... 123, 1786, 2663, 3288  
 \ifMT@inlist@ ..... 352,  
     397, 470, 640, 662, 679, 688, 708, 1332,  
     1410, 1421, 1509, 1756, 1780, 1844, 1856, 2170  
 \ifMT@kerning ..... 124, 1325, 3198, 3225, 3283  
 \ifMT@noligatures ..... 124, 1365  
 \ifMT@nonselected ..... 1010, 1522, 1530  
 \ifMT@noreset ..... 1584, 1591  
 \ifMT@opt@auto ..... 2670, 3114  
 \ifMT@opt@DVI ..... 2670, 3057  
 \ifMT@opt@expansion ..... 2670, 3052  
 \ifMT@protrusion ..... 124, 493, 515, 3058  
 \ifMT@selected ..... 124, 3139, 3159  
 \ifMT@spacing ..... 124, 3183, 3217, 3237  
 ifpdf (package) ..... 7, 96  
 inputenc (package) .....  
     ..... 12, 19, 21, 38, 60, 62, 63, 65, 155, 157  
 \inputencoding ..... 943, 946
- J**
- jurabib (package) ..... 66
- K**
- kerning (option) ..... 88  
 keyval (package) ..... 27, 43, 71, 103  
 \knaccode ..... 1274, 1275, 1287, 1291, 1298  
 \knbccode ..... 1267, 1268, 1286, 1290, 1297  
 \knbscode ..... 1159, 1160, 1185, 1190, 1198
- L**
- ledmac (package) ..... 20, 22, 36, 66, 155  
 ledpar (package) ..... 20  
 \leftmargin kern ..... 501  
 letterspacing (option) ..... 90, 157  
 lineno (package) ..... 36  
 listings (package) ..... 18, 38, 57, 156  
 lmodern (package) ..... 6, 114  
 \LoadMicrotypeFile ..... 10, 16, 34, 2166, 3047  
 \lpcode ..... 736, 774, 775, 956, 960  
 \lsstyle ..... 17, 558, 579, 1323, 1360
- M**
- marvosym (package) .... 15, 16, 21, 24, 107, 145, 157  
 mathdesign (package) ..... 16, 103, 156  
 memoir (package) ..... 17, 66  
 \MicrotypeHook ..... 92, 2889  
 \microtypecontext .....  
     ..... 12, 16, 1346, 1814, 3049, 3263, 3266  
 \microtypesetup ..... 8, 11, 2893, 3048  
 minimal (package) ..... 38  
 \MT@font 193, 587, 663, 666, 745, 833, 840, 1464, 1534  
 \MT@abbr@ex ..... 441  
 \MT@abbr@ex@c ..... 441  
 \MT@abbr@ex@inh ..... 441  
 \MT@abbr@kn ..... 448  
 \MT@abbr@kn@c ..... 448  
 \MT@abbr@kn@inh ..... 448  
 \MT@abbr@nl ..... 441  
 \MT@abbr@pr ..... 441  
 \MT@abbr@pr@c ..... 441  
 \MT@abbr@pr@inh ..... 441  
 \MT@abbr@sp ..... 448  
 \MT@abbr@sp@c ..... 448  
 \MT@abbr@sp@inh ..... 448  
 \MT@active@features 1774, 1777, 1789, 1796, 1832,  
     1843, 2111, 3059, 3154, 3184, 3197, 3254, 3255  
 \MT@addto@setup 479, 536, 1732, 1814, 1815, 1933,  
     2040, 2781, 2894, 3024, 3182, 3216, 3253, 3269  
 \MT@auto ..... 1019, 3110, 3123, 3132, 3136  
 \MT@auto@ ..... 1007, 1019, 1023, 1081  
 \MT@autofalse ..... 126, 3121, 3131  
 \MT@autotruer ..... 126, 2861  
 \MT@babel@false ..... 133, 3034  
 \MT@babel@true ..... 133  
 \MT@begin@catcodes .... 568, 1411, 1412, 1435,  
     2174, 2194, 2208, 2231, 2245, 2432, 2874, 2875  
 \MT@cat ..... 939, 1726, 1728  
 \MT@catcodes ..... 100, 119  
 \MT@char 764, 774, 775, 780, 781, 784, 786, 801,  
     806, 831, 956, 957, 959–961, 1051, 1063,  
     1064, 1066, 1068, 1108, 1110, 1111, 1148,  
     1159, 1160, 1166, 1167, 1173, 1174, 1177,  
     1179, 1185–1187, 1189–1192, 1256, 1267,  
     1268, 1274, 1275, 1278, 1280, 1286, 1287,  
     1289–1291, 1564, 1580, 1581, 1586, 1656,  
     1658, 1660, 1686, 1692, 2492, 2493, 2498, 2501  
 \MT@char@ ..... 1564, 1570, 1575, 1580,  
     1595, 1597, 1603, 1604, 1606, 1619, 1620,  
     1623, 1624, 1627, 1628, 1632, 1634, 1659, 1700  
 \MT@charstring ..... 1577, 1658, 1663  
 \MT@check@font ..... 1755, 1771, 1834  
 \MT@check@font@cx ..... 1775, 1834  
 \MT@check@range ..... 2634, 2636  
 \MT@check@range@ ..... 2636, 2637  
 \MT@check@rlist ..... 2576, 2627  
 \MT@check@rlist@ ..... 2627, 2628  
 \MT@checklist@ ..... 619, 632, 1370  
 \MT@checklist@family ..... 672  
 \MT@checklist@font ..... 651  
 \MT@checklist@size ..... 701  
 \MT@checksetup ..... 2904, 2915, 2922, 2943, 2965  
 \MT@clist@break ..... 326, 646, 664, 696, 714  
 \MT@clist@function ..... 326  
 \MT@cnt@encoding ..... 2513, 2520, 2521  
 \MT@cnt@family ..... 2518, 2527, 2528  
 \MT@cnt@series ..... 2525, 2534, 2535  
 \MT@cnt@shape ..... 2532, 2540, 2541  
 \MT@config@file 2864, 2872, 2873, 2877, 2878, 2881  
 \MT@context ..... 1485, 1497, 1516, 1541  
 \MT@count ..... 277, 278, 414, 793,  
     801, 802, 807, 808, 813, 814, 818, 826, 863, 925  
 \MT@curr@file ..... 1415, 1416,  
     1423, 1424, 2176, 2177, 2336, 2454, 2877, 3302  
 \MT@curr@list@name ..... 824, 866, 949,  
     1077, 1396, 1554, 1695, 1703, 1708, 1716, 1722  
 \MT@curr@set@name ..... 1901, 1903–  
     1905, 1907, 1908, 1921, 1926, 1931, 1934,  
     1958, 1962, 1991, 2022, 2028, 2033, 2038,

- 2041, 2286, 2305, 2343, 2353, 2357, 2367,  
2378, 2381, 2384, 2401, 2417, 2423, 2455, 2457  
\MT@declare@char@inh ..... 2437, 2442, 2452  
\MT@declare@sets ..... 1888, 1894, 1902, 2113  
\MT@DeclareSet ..... 1875, 1876, 1878  
\MT@DeclareSet@ ..... 1879, 1882, 1885  
\MT@DeclareSetAndUseIt ..... 1873, 1874, 1878  
\MT@def@bool@opt .... 2747, 2763, 2767, 2776, 2785  
\MT@def@n ..... 186, 1867, 1868  
\MT@def@num@opt ..... 2811, 2826  
\MT@def@simple@bool@opt ..... 2763, 2766  
\MT@default@ex@set ..... 2140, 3163  
\MT@default@kn@set ..... 2140, 3205  
\MT@default@pr@set ..... 2140, 3068  
\MT@default@sp@set ..... 2140, 3190  
\MT@define@code@key ..... 2268, 2327–2330, 2477–2479  
\MT@define@code@key@font ..... 2291, 2332, 2481  
\MT@define@code@key@size ..... 2279, 2331, 2480  
\MT@define@context ..... 1839, 1865  
\MT@define@ex@opt@key ..... 2397, 2410–2412  
\MT@define@key@unit ..... 2375, 2394, 2395  
\MT@define@opt@key ..... 2345–2348, 2351  
\MT@define@opt@keys ..... 2333, 2355  
\MT@define@option ..... 2673, 2709, 2710  
\MT@define@option@ ..... 2716, 2744, 2745  
\MT@define@optionX ..... 2893, 2953, 2954  
\MT@define@optionX@ ..... 2956, 2986, 2987  
\MT@define@set@key@ ..... 1912–1915, 1919  
\MT@define@set@key@font ..... 1917, 2026  
\MT@define@set@key@size ..... 1916, 1951  
\MT@define@set@keys ..... 1911, 2077, 2078  
\MT@detokenize@c ..... 229, 1572, 1656  
\MT@detokenize@n ..... 229, 1689, 1690  
\MT@dimen@six ..... 742, 794, 818, 863  
\MT@dinfo ..... 46  
\MT@dinfo@nl ..... 46  
\MT@do@font ..... 413, 735, 1029, 1197, 1296  
\MT@documentfalse ..... 165  
\MT@documenttrue ..... 165, 3257  
\MT@dofalse .... 612, 623, 645, 667, 684, 713, 1374  
\MT@dotrue .... 612, 615, 642, 681, 690, 710, 1366  
\MT@draftfalse ..... 129, 2787  
\MT@drafttrue ..... 129, 2789  
\MT@edef@n ..... 188, 1521, 1545,  
1860, 2274, 2336, 2338, 2453, 2468, 2607, 2817  
\MT@encoding ..... 602,  
1480, 1492, 1572, 1656, 1688, 1707, 1714, 1722  
\MT@encoding@check ..... 2466, 2470  
\MT@end@catcodes ..... 1413,  
1452, 2182, 2205, 2227, 2242, 2256, 2447, 2876  
\MT@error ..... 36, 473, 483, 949  
\MT@ex@cc@name ..... 1003,  
1005, 1083, 1084, 1094, 1100, 1101, 1115, 2211  
\MT@ex@context ..... 1826, 1866  
\MT@ex@do ..... 1006, 1041  
\MT@ex@doc@contexts ..... 1866  
\MT@ex@factor ..... 135, 1020, 1087  
\MT@ex@factor@ 1020, 1031, 1034, 1053, 1054, 1081  
\MT@ex@inh@name ..... 1065, 1066, 1068  
\MT@ex@level ..... 135, 3155, 3157  
\MT@ex@max ..... 150, 1056, 1057  
\MT@ex@min ..... 150, 1059, 1060  
\MT@ex@setname ..... 2095, 3160, 3161, 3163, 3164  
\MT@ex@split ..... 1043, 1047  
\MT@exp@one@n 194, 332, 469, 590, 592, 637, 656,  
1420, 1771, 1778, 1810, 1843, 1854, 1866,  
1894, 1933, 2040, 2088, 2133, 2169, 2442, 3263  
\MT@exp@two@c ..... 195,  
220, 224, 235, 366, 589, 1571, 1576, 1657  
\MT@exp@two@n .... 196, 677, 686, 1328, 1925, 2032  
\MT@expansion ..... 595, 996, 3175  
\MT@expansionfalse ..... 125, 3030, 3053, 3149  
\MT@expansiontrue ..... 125, 2860  
\MT@extra@context 1866, 2197, 2211, 2234, 2248,  
2296, 2300, 2301, 2304, 2349, 2428, 2573,  
2579, 2583, 2587, 2590, 2595, 2596, 2599, 2633  
\MT@extra@inputenc ..... 2427, 2451, 2456, 2457  
\MT@factor@default ..... 160, 2829, 2835, 3062  
\MT@family ..... 590, 602, 678, 1481, 2161, 2162  
\MT@familyalias ..... 591,  
592, 608, 685, 687, 689, 691, 1491, 1493, 2163  
\MT@feat .... 612, 747, 845, 847, 849, 850, 852,  
853, 859, 861, 864, 872–876, 878, 880–884,  
887, 888, 893, 897, 900, 903, 906–908, 917,  
918, 923, 942, 945, 947, 1386, 1388, 1391,  
1392, 1396, 1398, 1399, 1401, 1517, 1519,  
1521, 1525, 1529, 1533, 1536, 1543, 1545,  
1547, 1550, 1557, 1561, 1696, 1697, 1726, 1728  
\MT@features ..... 462,  
1866, 1888, 2077, 2082, 2127, 2355, 2437, 2885  
\MT@features@long ..... 462, 469, 474, 1865, 2449  
\MT@file@list ..... 1407,  
1409, 1417, 1420, 1425, 1428, 2169, 2173  
\MT@find@file ..... 590, 592, 1407  
\MT@fix@font@spec ..... 1930, 1947, 2037  
\MT@font ..... 193,  
589, 660, 736, 737, 743, 750, 774, 775,  
780, 781, 801, 806, 821, 826, 831, 931, 934,  
956, 957, 960, 961, 1007, 1023, 1029, 1063,  
1064, 1108, 1111, 1159, 1160, 1166, 1167,  
1173, 1174, 1185–1187, 1190–1192, 1198–  
1200, 1267, 1268, 1274, 1275, 1286, 1287,  
1290, 1291, 1297, 1298, 1377, 1731, 1748,  
1749, 1751, 1760, 1771, 1773, 1778, 1800, 1811  
\MT@font@list ..... 1730, 1771–1773, 1833  
\MT@gdef@n ..... 186, 2189,  
2204, 2226, 2241, 2255, 2401, 2417, 2459, 2887  
\MT@get@axis ..... 2048–2051, 2064, 2313–2316  
\MT@get@basefamily ..... 1419, 1453  
\MT@get@char@unit ..... 765, 895, 921, 1149, 1257  
\MT@get@charwd ..... 799, 818, 898, 921  
\MT@get@config ..... 2864  
\MT@get@dimen@six ..... 722, 742, 1127, 1235  
\MT@get@ex@opt ..... 999, 1014, 1081  
\MT@get@ex@opt@ ..... 1089–1091, 1093, 1099  
\MT@get@font ..... 2031, 2044  
\MT@get@font@and@size ..... 2295, 2309  
\MT@get@font@dimen ..... 820, 901  
\MT@get@highlevel ... 1924, 1937, 2066, 2273, 2467  
\MT@get@inh@list ..... 725, 1001, 1130, 1238, 1541  
\MT@get@listname ..... 1463, 1519, 1543  
\MT@get@listname@ ..... 1463  
\MT@get@opt ..... 723, 870, 1128, 1236  
\MT@get@range ..... 1955, 1965, 2283  
\MT@get@size .. 1972, 1977, 1986, 2005, 2058, 2324  
\MT@get@slot 763, 1050, 1147, 1255, 1564, 2491, 2497



- `\MT@get@space@unit` ..... 895,  
     986, 1157, 1164, 1171, 1218, 1222, 1265, 1272  
`\MT@get@unit` ..... 903, 911  
`\MT@get@unit@` ..... 911  
`\MT@glet` ..... 185, 480, 1353, 1760, 1834–  
     1837, 2121, 2163, 2615, 3068, 3163, 3190, 3205  
`\MT@gobble@settings` ..... 742  
`\MT@if@false` ..... 123, 1783, 2631, 3284  
`\MT@if@list@exists` 721, 998, 1013, 1126, 1234, 1516  
`\MT@if@true` .....  
     123, 1776, 2640, 2644, 2652, 2657, 3285–3287  
`\MT@ifdefined@c@T` ..... 200, 527, 542, 783,  
     1065, 1176, 1277, 2161, 2456, 2562, 2889, 2892  
`\MT@ifdefined@c@TF` ..... 200, 494, 943,  
     1343, 1520, 1544, 2572, 3065, 3160, 3187, 3202  
`\MT@ifdefined@n@T` ..... 200, 634,  
     653, 674, 703, 784, 906, 941, 1066, 1094,  
     1177, 1278, 1499, 1725, 1904, 2156, 2214,  
     2339, 2521, 2528, 2535, 2541, 2573, 2587, 3008  
`\MT@ifdefined@n@TF` ..... 200, 608,  
     617, 635, 654, 675, 704, 872, 880, 1083,  
     1100, 1368, 1391, 1487, 1572, 2097, 2101,  
     2142, 2548, 2606, 2622, 2692, 2729, 2886, 3261  
`\MT@ifdim` .. 286, 386, 387, 391, 392, 1988, 1997,  
     2638–2640, 2642, 2643, 2650–2652, 2655, 2656  
`\MT@ifdimen` ..... 265, 2017, 2366, 2383, 2849  
`\MT@ifempty` ..... 237,  
     772, 778, 1156, 1163, 1170, 1264, 1271,  
     1357, 1845, 1887, 1892, 1939, 1945, 1966,  
     1967, 1980, 1981, 2045, 2052, 2067, 2081,  
     2086, 2126, 2131, 2310, 2318, 2335, 2349,  
     2352, 2399, 2436, 2441, 2471, 2678, 2721,  
     2749, 2794, 2813, 2828, 2839, 2866, 2901, 2962  
`\MT@ifint` ..... 248, 1626, 2400, 2816, 2831  
`\MT@ifstreq` ..... 293, 1314, 1387,  
     1458–1460, 1518, 1592, 1809, 2359, 2365,  
     2380, 2382, 2681, 2683, 2686, 2689, 2724,  
     2726, 2752, 2753, 2797, 2798, 2803, 2842,  
     2848, 2903, 2910, 2914, 2921, 2964, 2970, 3240  
`\MT@in@clist` ..... 352, 469, 637, 657, 677,  
     686, 1328, 1409, 1420, 1771, 1778, 1843, 2169  
`\MT@in@rlist` ..... 380, 706, 1508  
`\MT@in@rlist@` ..... 380  
`\MT@in@rlist@` ..... 380  
`\MT@in@tlist` ..... 368, 1854  
`\MT@in@tlist@` ..... 368  
`\MT@increment` ..... 414, 2520, 2527, 2534, 2540  
`\MT@info` ..... 36, 50, 587, 2099,  
     2906, 2912, 2918, 2925, 2967, 2972, 2996, 3000  
`\MT@info@missing@char` ..... 802, 808, 829  
`\MT@info@n1` 36, 47, 51, 495, 586, 831, 838, 2793,  
     2873, 3056, 3061, 3066, 3069, 3073, 3156,  
     3161, 3164, 3176, 3186, 3188, 3191, 3195,  
     3201, 3203, 3206, 3209, 3221, 3229, 3272, 3289  
`\MT@inh@do` ..... 1556, 2482  
`\MT@inh@feat` ..... 2427, 2436, 2439, 2450  
`\MT@inh@split` ..... 2484, 2488  
`\MT@inlist@false` ..... 352, 356, 369, 381, 1786  
`\MT@inlist@true` ..... 352, 358, 376, 388, 393, 1786  
`\MT@is@active` ..... 1571, 1639  
`\MT@is@char` ..... 1576, 1657, 1663  
`\MT@is@composite` ..... 1574, 1684  
`\MT@is@feature` ..... 468, 1893, 2087, 2132  
`\MT@is@letter` ..... 1569, 1593, 1660, 1692  
`\MT@is@number` ..... 1609, 1614  
`\MT@is@symbol` ..... 1573, 1655  
`\MT@iterate` ..... 402  
`\MT@kerning` ..... 598, 1228, 3232  
`\MT@kerning@false` ..... 132, 3033, 3231  
`\MT@kerning@true` ..... 132, 1339  
`\MT@kn@cc@name` ..... 1240, 1242, 1305, 2248  
`\MT@kn@context` ..... 1314, 1827, 1866  
`\MT@kn@do` ..... 1243, 1246  
`\MT@kn@doc@contexts` ..... 1866  
`\MT@kn@factor` ..... 135  
`\MT@kn@factor@` ..... 872, 1315, 1317  
`\MT@kn@inh@name` ..... 1277, 1278, 1280  
`\MT@kn@max` ..... 150  
`\MT@kn@min` ..... 150  
`\MT@kn@setname` .....  
     .. 1331, 1334, 1342, 3202, 3203, 3205, 3206  
`\MT@kn@split` ..... 1248, 1252  
`\MT@kn@split@val` ..... 1258, 1262  
`\MT@kn@unit` ..... 135  
`\MT@kn@unit@` ..... 880, 1308  
`\MT@led@kern` ..... 491  
`\MT@led@unhbox@line` ..... 491  
`\MT@led@mac@setup` ..... 491, 538  
`\MT@let@cn` ..... 191, 192,  
     609, 729, 1005, 1084, 1134, 1242, 1386,  
     1398, 1513, 1517, 1550, 2009, 2343, 2455, 2814  
`\MT@let@nc` ..... 190,  
     908, 1529, 1557, 1561, 1781, 1833, 1907,  
     1921, 2028, 2219, 2357, 2367, 2378, 2381,  
     2384, 2423, 2608, 2620, 2687, 2690, 2917, 2924  
`\MT@let@nn` 192, 873, 878, 881, 893, 1101, 1104, 1853  
`\MT@letterspacing` ..... 135, 1344, 3211, 3212  
`\MT@letterspacing@` ..... 1316, 1343, 1344, 1356  
`\MT@letterspacing@default` ..... 164, 3212  
`\MT@listname` .....  
     1463, 1510, 1513, 1520, 1521, 1523, 1525,  
     1544, 1545, 1548, 1550, 1554, 1557, 2500, 2507  
`\MT@load@list` ..... 727, 1003, 1132, 1240, 1384  
`\MT@loop` ..... 402, 409, 2619  
`\MT@lower` ..... 1959, 1965, 2286  
`\MT@make@string` ..... 440, 1949, 2155, 2168  
`\MT@map@clist@` ..... 326  
`\MT@map@clist@c` ..... 326, 1777, 1789, 1796,  
     1832, 1865, 1888, 1890, 2077, 2082, 2084,  
     2127, 2129, 2355, 2437, 2439, 2449, 2885, 3006  
`\MT@map@clist@n` ..... 326,  
     616, 1367, 1866, 1922, 1953, 2029, 2187,  
     2271, 2281, 2293, 2494, 2676, 2719, 2898, 2959  
`\MT@map@tlist@` ..... 342  
`\MT@map@tlist@c` ..... 342, 371, 382,  
     785, 1067, 1178, 1279, 1467, 1802, 2574, 2632  
`\MT@map@tlist@n` ..... 342, 1447, 2764, 2824  
`\MT@maybe@do` ..... 612, 719, 996, 1121, 1229  
`\MT@MT` ..... 35, 36, 37, 40, 42,  
     43, 95, 484, 486, 2867, 2996, 3000, 3057, 3244  
`\MT@next@listname` ..... 1470, 1479  
`\MT@next@listname@` ..... 1488, 1500, 1506  
`\MT@nl@setname` ..... 2114  
`\MT@noligatures` ..... 600, 1363  
`\MT@noligatures@false` ..... 128  
`\MT@noligatures@true` ..... 128, 2112

- `\MT@nonselectedfalse` ..... 1010, 1024  
`\MT@nonselectedtrue` ..... 1010, 1012  
`\MT@noresetfalse` ..... 1591, 1592, 1599, 1608, 1629  
`\MT@noresettrue` ..... 1567, 1591, 1617  
`\MT@old@cmd` ..... 26, 31–34  
`\MT@opt@autofalse` ..... 2671  
`\MT@opt@autotrue` ..... 2671  
`\MT@opt@DVifalse` ..... 2672  
`\MT@opt@DVitrue` ..... 2672, 2769, 2772  
`\MT@opt@expansionfalse` ..... 2670  
`\MT@opt@expansiontrue` ..... 2670  
`\MT@options` ..... 2259  
`\MT@orig@add@accent` ..... 1764  
`\MT@orig@foreign@language` ..... 3278, 3280  
`\MT@orig@pickupfont` ..... 556, 1736, 1766  
`\MT@orig@select@language` ..... 3273, 3275  
`\MT@pdfcprot@error` ..... 482, 537  
`\MT@pdftex@no` ..... 53, 87, 91, 179, 3135  
`\MT@permute` ... 2203, 2225, 2240, 2254, 2462, 2512  
`\MT@permute@` ..... 2512  
`\MT@permute@@` ..... 2512  
`\MT@permute@@@` ..... 2512  
`\MT@permute@@@@` ..... 2512  
`\MT@permute@@@@@` ..... 2539, 2544  
`\MT@permute@@@@@@` ..... 2552, 2554  
`\MT@permute@define` ..... 2545, 2555–2557, 2604  
`\MT@permute@reset` ..... 2515, 2610  
`\MT@permute@reset@` ..... 2611–2614, 2617  
`\MT@permutelists` .. 2197, 2211, 2234, 2248, 2296,  
2297, 2304, 2460, 2568, 2569, 2573, 2579,  
2583, 2587–2590, 2599, 2600, 2633, 2664, 2665  
`\MT@pickupfont` ..... 1763, 1768  
`\MT@pr@c@name` ..... 727, 729, 966, 2197  
`\MT@pr@context` ..... 1826, 1866  
`\MT@pr@do` ..... 730, 754  
`\MT@pr@doc@contexts` ..... 1866  
`\MT@pr@factor` ..... 135, 2832, 3062, 3063  
`\MT@pr@factor@` ..... 872  
`\MT@pr@inh@name` ..... 783, 784, 786  
`\MT@pr@level` ..... 135, 3060, 3061  
`\MT@pr@max` ..... 150  
`\MT@pr@min` ..... 150  
`\MT@pr@setname` ..... 2095, 3065, 3066, 3068, 3069  
`\MT@pr@split` ..... 756, 760  
`\MT@pr@split@val` ..... 766, 770  
`\MT@pr@unit` ..... 135, 2850, 3064  
`\MT@pr@unit@` ..... 880, 969  
`\MT@preset@aux` ..... 971,  
973, 975, 976, 979, 1310, 1312, 1319, 1320  
`\MT@preset@aux@factor` .. 971, 979, 1212–1214, 1310  
`\MT@preset@aux@space` ..... 973, 979, 1216, 1312  
`\MT@preset@ex` ..... 1095, 1114  
`\MT@preset@kn` ..... 1303  
`\MT@preset@kn@` ..... 1303  
`\MT@preset@pr` ..... 964  
`\MT@preset@pr@` ..... 964  
`\MT@preset@sp` ..... 1205  
`\MT@preset@sp@` ..... 1205  
`\MT@ProcessOptionsWithKV` ..... 3003, 3023  
`\MT@protrusion` ..... 594, 719, 3072  
`\MT@protrusionfalse` ..... 124, 3029  
`\MT@protrusiontrue` ..... 124, 2857  
`\MT@rbba@expansion` ..... 456  
`\MT@rbba@kerning` ..... 456  
`\MT@rbba@protrusion` ..... 456  
`\MT@rbba@spacing` ..... 456  
`\MT@register@font` ..... 1758, 1773, 1835  
`\MT@register@font@cx` ..... 1795, 1835  
`\MT@register@subst@font` ..... 1752, 1772, 1836  
`\MT@register@subst@font@cx` ..... 1788, 1836  
`\MT@rem@from@clist` ..... 363, 1810, 2263  
`\MT@rem@from@lists` ..... 1804, 1808  
`\MT@repeat` ..... 402, 411, 2625  
`\MT@requires@etex` 166, 200, 229, 415, 424, 799, 829  
`\MT@requires@pdftex` ..... 178, 248,  
265, 293, 491, 791, 1032, 1120, 1228, 1323,  
1363, 2109, 2416, 2422, 2859, 3112, 3181, 3235  
`\MT@reset@context` ..... 1818, 1821, 1824, 1852  
`\MT@reset@context@` ..... 1824, 1852  
`\MT@reset@ef@codes` ..... 1000, 1022, 1032, 1096  
`\MT@reset@ef@codes@` ..... 1027, 1035, 1039  
`\MT@reset@kn@codes` ..... 1237, 1244, 1294  
`\MT@reset@kn@codes@` ..... 1294  
`\MT@reset@pr@codes` ..... 724, 731, 740  
`\MT@reset@pr@codes@` ..... 740  
`\MT@reset@sp@codes` ..... 1129, 1136, 1195  
`\MT@reset@sp@codes@` ..... 1195  
`\MT@restore@catcodes` ..... 112, 120  
`\MT@restore@p@h` ..... 524, 532, 533, 554, 564  
`\MT@saved@setupfont` ..... 2994, 3001  
`\MT@scale` ..... 424, 794, 846, 860, 863, 1054, 1316  
`\MT@scale@factor` ..... 796, 815, 844, 981, 1116  
`\MT@scale@to@em` ..... 773, 779, 791, 812,  
987, 1158, 1165, 1172, 1219, 1223, 1266, 1273  
`\MT@selectedfalse` ..... 127  
`\MT@selectedtrue` ..... 127  
`\MT@series` ..... 602, 1482, 1494  
`\MT@set@all@ex` ..... 1027, 1117  
`\MT@set@all@kn` ..... 1294, 1321  
`\MT@set@all@pr` ..... 733, 740, 977  
`\MT@set@all@sp` ..... 1195, 1226  
`\MT@set@babel@context` ..... 3259, 3276, 3281, 3293  
`\MT@set@default@set` ..... 2127, 2133, 2140  
`\MT@set@ex@codes` ..... 1026, 3140, 3142  
`\MT@set@ex@codes@n` ..... 1010, 1026, 3142  
`\MT@set@ex@codes@s` ..... 997, 3140  
`\MT@set@ex@heirs` ..... 1069, 1107  
`\MT@set@inh@list` ..... 2433, 2435  
`\MT@set@inputenc` . 726, 939, 1002, 1131, 1239, 1555  
`\MT@set@kn@codes` ..... 1233  
`\MT@set@kn@heirs` ..... 1281, 1285  
`\MT@set@listname` .....  
.... 728, 871, 1004, 1082, 1133, 1241, 1695  
`\MT@set@named@keys` .. 2199, 2213, 2236, 2250, 2259  
`\MT@set@pr@codes` ..... 720  
`\MT@set@pr@heirs` ..... 787, 955  
`\MT@set@sp@codes` ..... 1125  
`\MT@set@sp@heirs` ..... 1180, 1184  
`\MT@SetExpansion` ..... 2209, 2211  
`\MT@SetExtraKerning` ..... 2246, 2248  
`\MT@SetExtraSpacing` ..... 2232, 2234  
`\MT@SetProtrusion` ..... 2195, 2197  
`\MT@setup@` ..... 478, 479, 480, 2779, 2780  
`\MT@setup@contexts` ..... 1817, 1831  
`\MT@setupfont` .. 585, 1757, 2994, 2997, 3001, 3036

- \MT@setupfont@hook ..... 525, 539, 541, 546, 554, 564, 573, 588  
 \MT@shape ..... 602, 1483, 1495  
 \MT@shrink ..... 135, 1017, 3079,  
     3081, 3083, 3087, 3088, 3091, 3095, 3145, 3158  
 \MT@shrink@ ..... 1007, 1017, 1023, 1081  
 \MT@shrink@default ..... 161, 3083  
 \MT@size ..... 387, 391, 392, 602, 709, 712  
 \MT@size@name ..... 380, 1510  
 \MT@sp@cc@name ..... 1132, 1134, 1207, 2234  
 \MT@sp@context ..... 1827, 1866, 3240  
 \MT@sp@do ..... 1135, 1138  
 \MT@sp@doc@contexts ..... 1866  
 \MT@sp@factor ..... 135  
 \MT@sp@factor@ ..... 872  
 \MT@sp@inh@name ..... 1176, 1177, 1179  
 \MT@sp@max ..... 150  
 \MT@sp@min ..... 150  
 \MT@sp@setname ..... 3187, 3188, 3190, 3191  
 \MT@sp@split ..... 1140, 1144  
 \MT@sp@split@val ..... 1150, 1154  
 \MT@sp@unit ..... 135  
 \MT@sp@unit@ ..... 880, 1210  
 \MT@spacing ..... 597, 1120, 3194, 3224  
 \MT@spacingfalse ..... 131, 3032, 3223  
 \MT@spacingtrue ..... 131  
 \MT@split@name ..... 589, 602  
 \MT@split@val ..... 1154  
 \MT@step ..... 135, 1018, 3086, 3102, 3109, 3159  
 \MT@step@ ..... 1007, 1018, 1023, 1081  
 \MT@step@default ..... 161  
 \MT@stretch ..... 135, 1016, 3076, 3077,  
     3080, 3081, 3087, 3089, 3094, 3097, 3144, 3158  
 \MT@stretch@ ..... 1007, 1016, 1023, 1081  
 \MT@stretch@default ..... 161, 3077  
 \MT@temp ..... 2047, 2075, 2312  
 \MT@tempencoding ..... 2547, 2548, 2558, 2565  
 \MT@tempfamily ..... 2559  
 \MT@tempseries ..... 2560  
 \MT@tempshape ..... 2561  
 \MT@tempsize ..... 2562, 2572, 2575, 2580, 2615  
 \MT@test@ast ..... 1938, 1943  
 \MT@testrest ..... 1591  
 \MT@tlist@break ..... 342, 377, 399, 1472, 2667  
 \MT@toks ..... 122, 831, 839, 866,  
     1077, 1568, 1597, 1604, 1649, 1700, 1712, 1720  
 \MT@try@order ..... 1467, 1475  
 \MT@undefined@char ..... 1644, 1646, 1654  
 \MT@upper ..... 1959, 1965, 2286  
 \MT@use@set ..... 2082, 2088, 2095  
 \MT@val ..... 1845–1847, 1849,  
     1855, 1857, 1860, 1923, 1927, 1938, 1940,  
     1954–1956, 1968, 1971, 1973, 1976, 1978,  
     1979, 1985, 1987, 1988, 1990, 1993, 1995,  
     2006, 2009, 2013, 2017–2019, 2021, 2023,  
     2030, 2031, 2034, 2054, 2056, 2059–2061,  
     2065, 2067, 2069, 2071, 2272, 2274, 2282–  
     2284, 2294, 2295, 2299, 2305, 2320, 2322,  
     2465, 2466, 2468, 2472, 2493, 2500, 2506,  
     2507, 2677, 2678, 2681, 2683, 2686, 2689,  
     2692, 2693, 2698, 2720, 2721, 2724, 2726,  
     2729, 2730, 2735, 2899, 2901, 2903, 2910,  
     2914, 2921, 2929, 2960, 2962, 2964, 2970, 2975  
 \MT@vinfo ..... 36, 47,  
     586, 629, 875, 884, 888, 923, 1085, 1102,  
     1378, 1392, 1416, 1424, 1427, 1523, 1525,  
     1531, 1825, 1849, 2171, 2177, 2793, 2802, 3262  
 \MT@warn@ascii ..... 1606, 1699  
 \MT@warn@axis@empty ..... 2053, 2068, 2073, 2319  
 \MT@warn@code@too@large ..... 850, 853, 857  
 \MT@warn@err ..... 36, 2799, 2800  
 \MT@warn@ex@too@large ..... 1057, 1060, 1075  
 \MT@warn@maybe@inputenc ..... 1713, 1721, 1724  
 \MT@warn@number@too@large ..... 1633, 1705  
 \MT@warn@preset@towidth ..... 970, 990, 1211, 1309  
 \MT@warn@rest ..... 1585, 1710  
 \MT@warn@unknown ..... 1582, 1718  
 \MT@warning ..... 27, 36,  
     1350, 1388, 1401, 1533, 1905, 1989, 2021,  
     2074, 2104, 2118, 2146, 2157, 2179, 2340,  
     2360, 2369, 2386, 2403, 2419, 2473, 2566,  
     2588, 2664, 2799, 2843, 2889, 2928, 2947, 2974  
 \MT@warning@nl ..... 36, 90, 507, 516, 548,  
     744, 822, 864, 916, 991, 1076, 1700, 1706,  
     1711, 1719, 1738, 2216, 2697, 2734, 2754,  
     2800, 2804, 2818, 2833, 2852, 2880, 3026,  
     3104, 3115, 3126, 3146, 3218, 3226, 3241, 3295  
 \MT@while@num ..... 407, 413, 569–571  
 \MT@with@package ..... 481, 490,  
     529, 533, 537, 538, 544, 563, 567, 578, 1745  
 \MT@xadd ..... 312, 1333, 1417, 1425,  
     1428, 1790, 1798, 1857, 1957, 2173, 2285, 2499  
 \MT@xaddb ..... 319, 2303, 2578  
 \MT@xdef@n .. 188, 2098, 2102, 2144, 2149, 2160,  
     2296, 2353, 2457, 2599, 2693, 2695, 2730, 2732
- N**
- \normalfont ..... 584, 3168, 3172  
 \nullfont ..... 584
- P**
- \pdfadjustinterwordglue ..... 2986, 3185  
 \pdfadjustspacing ..... 2954, 3155, 3171  
 \pdfappendkern ..... 1340, 2988, 3200  
 \pdfcpot (package) ..... 5, 19, 20, 36, 43, 66, 153  
 \pdffontexpand ..... 1007, 1023  
 \pdfmatch ..... 250, 267  
 \pdfnoligatures ..... 1377  
 \pdfoutput ..... 57,  
     2769, 2770, 2772, 2773, 2858, 3051, 3056, 3113  
 \pdfprependkern ..... 1341, 2987, 3199  
 \pdfprotrudechars ..... 2953, 3060  
 \pdfstrcmp ..... 295  
 \pdftexrevision ..... 56, 61, 70, 72  
 \pdftexversion ..... 55, 59–61, 63, 65, 67, 69, 77  
 \pickup@font .....  
     556, 1737, 1739, 1744–1746, 1763, 1766, 1768  
 pifont (package) ..... 66  
 protrusion (option) ..... 5, 6, 10, 88  
 PSNFSS (package) ..... 148  
 pstricks (package) ..... 7, 96  
 pxfonts (package) ..... 16, 102
- Q**
- qfonts (package) ..... 16, 102, 156

R	
<code>resize (package)</code> .....	73
<code>\rightmarginkern</code> .....	500
<code>\rptcode</code> .....	737, 780, 781, 957, 961
S	
<code>selected (option)</code> .....	7, 13, 14, 22, 48, 59, 89, 154
<code>\selectfont</code> .....	1820, 1829, 3301
<code>\set@fontsize</code> .....	2013
<code>\SetExpansion</code> .....	7, 13, 2207, 3040
<code>\SetExtraKerning</code> .....	15, 2244, 3043
<code>\SetExtraSpacing</code> .....	14, 2230, 3042
<code>\SetProtrusion</code> .....	6, 11, 2193, 3039
<code>\sfcode</code> .....	3239
<code>\shbcode</code> .....	1173, 1174, 1187, 1192, 1200
<code>\showhyphens</code> .....	3166, 3170
<code>shrink (option)</code> .....	7, 14, 22, 90, 97, 156
<code>slantsc (package)</code> .....	127
<code>soul (package)</code> .....	38, 150, 156
<code>spacing (option)</code> .....	88
<code>\stbcode</code> .....	1166, 1167, 1186, 1191, 1199
<code>step (option)</code> .....	7, 22, 90, 97, 154, 156
<code>stretch (option)</code> .....	7, 14, 18, 22, 90, 97
T	
<code>tex4ht (package)</code> .....	37, 157
<code>\textls</code> .....	18, 559, 580, 1356
<code>tipa (package)</code> .....	66
<code>trace (package)</code> .....	67, 157
<code>\tracingmicrotype</code> .....	46
<code>txfonts (package)</code> .....	16, 102
U	
<code>ucs (package)</code> .....	19, 63
<code>unit (option)</code> .....	6, 22, 155, 156
<code>\UseMicrotypeSet</code> ...	3, 10, 22, 33, 1883, 2079, 3038
V	
<code>verbose (option)</code> .....	8, 22, 90, 155
Z	
<code>zefonts (package)</code> .....	16, 102

## C The L<sup>A</sup>T<sub>E</sub>X Project Public License

LPPL Version 1.3c 2006-05-20

Copyright 1999, 2002–2006 L<sup>A</sup>T<sub>E</sub>X3 Project

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

### Preamble

The L<sup>A</sup>T<sub>E</sub>X Project Public License (LPPL) is the primary license under which the the L<sup>A</sup>T<sub>E</sub>X kernel and the base L<sup>A</sup>T<sub>E</sub>X packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is T<sub>E</sub>X-related (such as a L<sup>A</sup>T<sub>E</sub>X package), but it is written in such a way that you can use it even if your work is unrelated to T<sub>E</sub>X.

The section ‘WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE’, below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the L<sup>A</sup>T<sub>E</sub>X3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document ‘cfigguide.tex’ and ‘modguide.tex’ in the base L<sup>A</sup>T<sub>E</sub>X distribution for suggestions.

### Definitions

In this license document the following terms are used:

**Work:** Any work being distributed under this License.

**Derived Work:** Any work that under any applicable law is derived from the Work.

**Modification:** Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

**Modify:** To apply any procedure that produces a Derived Work under any applicable law.

**Distribution:** Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun’s Network File System (NFS).

**Compiled Work:** A version of the Work that has been processed into a form where it is directly usable on a computer system. This processing may include using

installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

**Current Maintainer:** A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the ‘Copyright Holder’ under any applicable law.

**Base Interpreter:** A program or process that is normally

needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a ‘L<sup>A</sup>T<sub>E</sub>X-Format’ or in the case of files belonging to the ‘L<sup>A</sup>T<sub>E</sub>X-format’ a program implementing the ‘T<sub>E</sub>X language’.

## Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.
2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.
5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
  - (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
  - (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.
  - (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
  - (d) You distribute at least one of the following with the Derived Work:
    - i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
    - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or

nullify any sections of this license as they pertain to the results of applying that process.

10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
- (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow

each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

## No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any au-

thor named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

## Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
  - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.

- (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a L<sup>A</sup>T<sub>E</sub>X work, this could be done, for example, by posting to `comp.text.tex`.)

3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.
- (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.
4. If you make an ‘intention announcement’ as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.

5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a



prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b

above.

## Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

### Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document ‘modguide.tex’ in the base L<sup>A</sup>T<sub>E</sub>X distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing L<sup>A</sup>T<sub>E</sub>X under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L<sup>A</sup>T<sub>E</sub>X, the discussion in ‘modguide.tex’ may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

### A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

### How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
% http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status ‘maintained’.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the ‘Work’ referring to the three files ‘pig.dtx’, ‘pig.ins’, and ‘pig.sty’ (the last being generated from ‘pig.dtx’ using ‘pig.ins’), the ‘Base Interpreter’ referring to any ‘L<sup>A</sup>T<sub>E</sub>X-Format’, and both ‘Copyright Holder’ and ‘Current Maintainer’ referring to the person ‘M. Y. Name’.

If you do not want the Maintenance section of LPPL to apply to your Work, change ‘maintained’ above into ‘author-maintained’. However, we recommend that you use ‘maintained’ as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

### Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case (e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

### Important Recommendations

#### Defining What Constitutes the Work

The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.