



# 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)

v1.9e — 2006/07/28

## Abstract

The `microtype` package provides an interface to the micro-typographic extensions of pdf $\TeX$ : most prominently, character protrusion and font expansion, furthermore the possibility to disable all ligatures of a font.<sup>1</sup> It allows to apply these features to customizable sets of fonts, and to configure all micro-typographic aspects of the fonts in a straight-forward and flexible way. Settings for various fonts are provided.<sup>2</sup>

Note that font expansion and character protrusion will only work with pdf $\TeX$ , at least version 0.14f. Automatic font expansion requires version 1.20 or newer. Disabling ligatures require pdf $\TeX$  1.30. The package will by default enable the features that can safely be assumed to work.

This work may be distributed and/or modified under the conditions of the [L \$\TeX\$ S 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 – 41] [Expansion – 48] [Interword Space (Glue) – 50]	
	[Additional Kerning – 52] [Letter spacing – 54] [Disabling	
	Ligatures – 55] [Loading the Configuration – 56] [Translating Characters	
	into Slots – 60] [Hook into $\LaTeX$ 's font selection – 65] [Context-sensitive	
	Setup – 67]	
13.5	Configuration . . . . .	70
	[Font Sets – 70] [Interaction with babel – 76] [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 – 107] [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>157</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 pdf<sub>T</sub>E<sub>X</sub>

pdf<sub>T</sub>E<sub>X</sub>, the <sub>T</sub>E<sub>X</sub> 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  $\LaTeX$  user interface for quite some time. Then, the `pdfcprot` package was released, which allowed  $\LaTeX$  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 <sub>T</sub>E<sub>X</sub> and  $\LaTeX$  users: font expansion can now take place automatically. That is, pdf<sub>T</sub>E<sub>X</sub> 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> pdf<sub>T</sub>E<sub>X</sub> version 1.40 (available from <http://sarovar.org/projects/pdfstex/>) 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  $\LaTeX$  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 pdf $\TeX$  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 pdf $\TeX$  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 pdf $\TeX$ ):

```
\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 `pdfcpot` 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** *<integer>* 20

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

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

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

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

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

```
\usepackage{microtype}
```

**selected** true, false false

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

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

### 3.4 Miscellaneous Options

**DVIoutput** true, false false

pdfTeX is not only able to generate PDF output but can also spit out DVI files.<sup>6</sup> The latter can be ordered with the option `DVIoutput`, which will set `\pdfoutput` to zero.

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

When generating DVI files, font expansion has to be enabled explicitly. *Automatic* font expansion will not work because `dvips` (resp. the DVI viewer) is not able to generate the expanded fonts on the fly.

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

- draft** true, false false
- final** If the draft option is passed to the package, *all micro-typographic extensions will be disabled*. The draft and final options may also be inherited from the class options; of course, you can override them in the package options.
- verbose** true, false, errors false  
Information on the settings used for each font will be written into the log file if you enable the verbose option, which is disabled by default.  
When microtype encounters a problem that is not fatal (e. g., an unknown character in the settings, or non-existent settings), it will by default only issue a warning and try to continue. Loading the package with verbose=errors will turn all warnings into errors, so that you can be sure that no problem will go unnoticed.
- config** *<file name>* microtype  
Various settings for this package will be loaded from a main configuration file, by default microtype.cfg (see section 5.4). You can have a different configuration file loaded instead by specifying its name *without the extension*, e. g., config=mycrotype.

### 3.5 Changing Options Later

`\microtypesetup`  $\{ \langle \text{key} = \text{value list} \rangle \}$

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

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

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

## 4 Selecting Fonts for Micro-Typography

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

`\DeclareMicrotypeSet` [*<features>*]  $\{ \langle \text{set name} \rangle \} \{ \langle \text{set of fonts} \rangle \}$

`\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 'basic`text`' 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 'all`text`', 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 Ä are valid, provided the character is actually declared in both the input and the font encoding. You also have the possibility to declare lists of characters that should inherit protrusion or expansion factors (see section 5.3).

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 (`={\langle left \rangle, \langle right \rangle}`), possibly scaled by a factor. A `unit` setting will only be taken into account if it is not `=character`.

**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><sub>X</sub> prohibits the use of the same font with different expansion parameters. If you do not want to create a clone of the font setup (this would require duplicating the `tfm/vf` files under a new name, and writing new `fd` files and `map` entries), you could exploit a dirty trick and load a minimally larger font for the paragraph in question. E. g., for a document typeset in 10pt:<sup>9</sup>

<sup>9</sup> Note that the `\expandpar` command can only be applied to complete paragraphs. If you are using Computer Modern Roman, you have to load the `fix-cm` package to be able to select fonts in arbitrary sizes. Finally, the reason I suggest to use a larger font, and not a smaller one, is to prevent a different design size being selected.

```

\SetExpansion
  [ stretch = 30,
    shrink = 60,
    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 pdfTeX: it does not allow different expansion limits or steps (even of different fonts) within one paragraph. The factor option influences the expansion factors of all characters (in contrast to the overall stretchability) of the font. For instance, if you want the italic shape to be expanded less, you could declare:

```

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

```

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

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

If the selected option has been passed to the package (cf. section 3.3), and settings for a font don't exist, font expansion will not be applied to this font at all. Should the extraordinary situation arise that you want to employ selected expansion in general but that all characters of a particular font (set) should be expanded or shrunk by the same amount, you would have to declare an empty list for these fonts.

### 5.3 Character Inheritance

```

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

```

In most cases, accented characters should inherit the protrusion resp. expansion factors from the respective base character. For example, all of the characters 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  $\langle \text{base character} \rangle = \langle \text{list of inheriting characters} \rangle$ . 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- $\langle \text{font family} \rangle$ .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 { $\langle \text{font name} \rangle$ } { $\langle \text{alias font} \rangle$ }
```

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

*a* Incomplete

*b* Aliases: Latin Modern (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor)

*c* Alias: mathdesign/Charter (mdbch)

*d* Settings inherited from italic shape

*e* Alias: mathdesign/URW Garamond (mdugm)

*f* Aliases: pxfonts (pxr), qfonts/QuasiPalatino (qpl)

*g* Aliases: txfonts (txr), qfonts/QuasiTimes (qtm)

*h* 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>E<sub>X</sub> 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: pdf<sub>l</sub>atex: font ptmr8r cannot be expanded (not an included Type1 font) and the PDF viewer complains about a missing font, e. g., Adobe Reader: 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: pdf<sub>l</sub>atex (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 pdf<sub>T</sub>E<sub>X</sub> 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<sub>\_</sub>mem<sub>\_</sub>size)=65536].  
When applying micro-typographic enhancement to a large document with a lot of fonts, pdf<sub>T</sub>E<sub>X</sub> may be running out of memory. The memory can be increased by setting `pdf_mem_size` to a larger value (maximum 524 288). For te<sub>T</sub>E<sub>X</sub>-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 pdf<sub>T</sub>E<sub>X</sub>, 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 Gaille*, *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 pdf`cprot` 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*, December 4, 2005. (Available from CTAN at [/systems/pdftex/manual/](#); latest version at <http://sarovar.org/projects/pdftex/>)

L<sup>A</sup>T<sub>E</sub>X3 Project Team, *L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub> font selection*, November 27, 2005. (Available from CTAN at [/macros/latex/doc/fntguide.pdf](#))

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

## 12 Short History

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

### 1.9e (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)  
‘DVOutput’ 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 ( $\min(\text{stretch}, \text{shrink})/5$ ) (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 pdf $\TeX$ .

config: Surrounds all configuration modules.

cfg-t: Surrounds latin text configurations.

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

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

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

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

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

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

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

Contributed by *Harald Harders*.

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

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

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

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

euf: Settings for 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  $\langle *package \rangle$

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

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



```

7 \newcommand*\SetExpansion[3] [] {}
8 \newcommand*\DisableLigatures[1] {}
9 \newcommand*\DeclareCharacterInheritance[3] [] {}
10 \newcommand*\LoadMicrotypeFile[1] {}
11 \newcommand*\microtypesetup[1] {}
12 \newcommand*\microtypecontext[1] {}
13 <beta>
14 \newcommand*\SetExtraSpacing[3] [] {}
15 \newcommand*\SetExtraKerning[3] [] {}
16 \newcommand*\DeclareMicrotypeBabelHook[2] {}
17 \newcommand*\lstyle{}
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.

```

36 \def\MT@error{\PackageError\MT@MT}
37 \def\MT@warning{\PackageWarning\MT@MT}
38 \def\MT@warning@n1#1{\MT@warning{#1\@gobble}}
39 \def\MT@warn@err#1{\MT@error{#1}}{%
40   This error message appears because you loaded the \MT@MT'\MessageBreak
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 $\TeX$ 's features for which we provide an interface here haven't always been available, and some specifics have changed over time. Therefore, we have to test which pdf $\TeX$  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 $\TeX$ :

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

```
53 \let\MT@pdftex@no\z@
```

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

```

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

```

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

```

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

```

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

```

73         \let\MT@pdftex@no\@ne
74         \fi
75         \fi
76         \else
77         \ifnum\pdftexversion < 14
78         \let\MT@pdftex@no\@ne
79         \fi
80         \fi
81         \fi
82         \fi
83 (beta) \fi
84 \fi
85 \fi
86 (debug)\MT@info@n1{0}{pdftex no: \number\MT@pdftex@no}

```

If we are not using pdf $\TeX$  or in case it is too old, we disable everything and exit here.

```

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

```

Still there? Then we can begin:

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

```

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

```

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

```

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

```

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

```

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

```

```

\MT@toks We need a token register.
122 \newtoks\MT@toks

\ifMT@protrusion These are the global switches ...
\ifMT@expansion 123 \newif\ifMT@protrusion
\ifMT@auto 124 \newif\ifMT@expansion
125 \newif\ifMT@auto
\ifMT@selected 126 \newif\ifMT@selected
\ifMT@noligatures 127 \newif\ifMT@noligatures
\ifMT@draft 128 \newif\ifMT@draft
129 \beta
\ifMT@spacing 130 \newif\ifMT@spacing
\ifMT@kerning 131 \newif\ifMT@kerning
\ifMT@babel 132 \newif\ifMT@babel
133 /beta

\MT@pr@level ... and numbers.
\MT@pr@factor 134 \let\MT@pr@level\tw@
\MT@pr@unit 135 \let\MT@pr@factor\@m
\MT@ex@level 136 \let\MT@pr@unit\@empty
137 \let\MT@ex@level\tw@
\MT@ex@factor 138 \let\MT@ex@factor\@m
\MT@stretch 139 \let\MT@stretch\@m@ne
\MT@shrink 140 \let\MT@shrink \m@ne
\MT@step 141 \let\MT@step \m@ne
142 *beta
\MT@sp@factor 143 \let\MT@sp@factor\@m
144 \let\MT@kn@factor\@m
\MT@sp@unit
\MT@kn@factor Default unit for spacing settings is space, default unit for kerning is 1em.
145 \let\MT@sp@unit\m@ne
\MT@kn@unit 146 \def\MT@kn@unit{1em}
\MT@letterspacing 147 \let\MT@letterspacing\m@ne
148 /beta

\MT@pr@min Minimum and maximum values allowed by pdfTEX.
\MT@pr@max 149 \def\MT@pr@min{-\@m}
\MT@ex@min 150 \let\MT@pr@max\@m
\MT@ex@max 151 \let\MT@ex@min\z@
\MT@ex@max 152 \let\MT@ex@max\@m
\MT@sp@min 153 *beta
\MT@sp@max 154 \def\MT@sp@min{-\@m}
155 \let\MT@sp@max\@m
\MT@kn@min 156 \def\MT@kn@min{-\@m}
\MT@kn@max 157 \let\MT@kn@max\@m
158 /beta

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

\MT@stretch@default Default values for expansion.
\MT@shrink@default 160 \def\MT@stretch@default{20 }
\MT@step@default 161 \def\MT@shrink@default{20 }
162 \def\MT@step@default{4 }

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

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

```

## 13.2 Auxiliary macros

<code>\MT@requires@etex</code>	For definitions that depend on e- $\TeX$ features.
	<pre> 165 \expandafter\let\expandafter\MT@requires@etex 166   \ifcase 0% 167     \ifx\TeXversion\undefined 1\else 168       \ifx\TeXversion\relax 1\else 169         \ifcase\TeXversion 1\fi 170       \fi 171     \fi\space 172   \@firstoftwo 173 \else 174   \@secondoftwo 175 \fi 176 <i>(debug)</i>\MT@info@n1{0}{\MT@requires@etex}{\not }running etex </pre>
<code>\MT@requires@pdftex</code>	For definitions that depend on a particular pdf $\TeX$ version.
	<pre> 177 \def\MT@requires@pdftex#1{% 178   \ifnum\MT@pdftex@no&lt;#1\relax 179     \expandafter\@secondoftwo 180   \else 181     \expandafter\@firstoftwo 182   \fi 183 } </pre>
<code>\MT@def@n</code>	This is <code>\@namedef</code> .
	<pre> 184 \def\MT@def@n#1{\expandafter\def\csname #1\endcsname} </pre>
<code>\MT@edef@n</code>	Its expanding version.
	<pre> 185 \def\MT@edef@n#1{\expandafter\edef\csname #1\endcsname} </pre>
<code>\MT@let@nc</code>	<code>\let</code> a <code>\csname</code> sequence to a command.
	<pre> 186 \def\MT@let@nc#1{\expandafter\let\csname #1\endcsname} </pre>
<code>\MT@let@cn</code>	<code>\let</code> a command to a <code>\csname</code> sequence.
	<pre> 187 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname} </pre>
<code>\MT@let@nn</code>	<code>\let</code> a <code>\csname</code> sequence to a <code>\csname</code> sequence.
	<pre> 188 \def\MT@let@nn#1{\expandafter\MT@let@cn\csname #1\endcsname} </pre>
<code>\MT@exp@string</code>	Remove trailing space.
	<pre> 189 \def\MT@exp@string{\expandafter\string} </pre>
<code>\MT@exp@one@n</code>	Expand the second token once and enclose it in braces.
	<pre> 190 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}} </pre>
<code>\MT@exp@two@c</code>	Expand the next two tokens after $\langle\#1\rangle$ once.
	<pre> 191 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter} </pre>
<code>\MT@exp@two@n</code>	Expand the next two tokens after $\langle\#1\rangle$ once and enclose them in braces.
	<pre> 192 \def\MT@exp@two@n#1#2#3{% 193   \expandafter\expandafter\expandafter 194   #1\expandafter\expandafter\expandafter 195   {\expandafter#2\expandafter}\expandafter{#3}} </pre>
	You do not wonder why <code>\MT@exp@one@c</code> doesn't exist, do you?
<code>\MT@ifdefined@c@T</code>	Wrapper for testing whether command resp. <code>\csname</code> sequence is defined. If we
<code>\MT@ifdefined@c@TF</code>	are running e- $\TeX$ , we will use its primitives <code>\ifdefined</code> and <code>\ifcsname</code> , which
<code>\MT@ifdefined@n@T</code>	decreases memory use substantially.
<code>\MT@ifdefined@n@TF</code>	<pre> 196 \MT@requires@etex{ </pre>

```

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

```

`\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`

```

225 \MT@requires@etex{
226 \def\MT@detokenize@n#1{\detokenize\expandafter{\string#1}}
227 \def\MT@detokenize@c#1{\detokenize
228 \expandafter\expandafter\expandafter{\expandafter\string#1}}
229 }{
230 \def\MT@detokenize@n#1{\string#1}
231 \def\MT@detokenize@c#1{\MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty}
232 }

```

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

```

233 \begingroup
234 \catcode`\%=12
235 \catcode`\&=14
236 \gdef\MT@ifempty#1{&
237 \if %#1%&
238 \expandafter\@firstoftwo
239 \else
240 \expandafter\@secondoftwo
241 \fi
242 }
243 \endgroup

```

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

```

244 \MT@requires@pdfTeX6{
245 \def\MT@ifint#1{%
246 \ifcase\pdfmatch{^[0-9]+ *$}\{#1}\relax
247 \expandafter\@secondoftwo
248 \else
249 \expandafter\@firstoftwo

```

```

250   \fi
251 }
252 }{
253   \def\MT@ifint#1{%
254     \if!\ifnum9<#1!\else?\fi
255     \expandafter\@firstoftwo
256   \else
257     \expandafter\@secondoftwo
258   \fi
259 }
260 }

```

`\MT@ifdimen` Test whether argument is dimension (or number). (nd and nc are new Didot resp. Cicero, added in pdf<sub>T</sub><sub>E</sub><sub>X</sub> 1.30; px is a pixel, it seems.)

```

261 \MT@requires@pdftex6{
262   \def\MT@ifdimen#1{%
263     \ifcase\pdfmatch{^[0-9]+([.][0-9]+)?|([.][0-9]+)%
264               (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
265     \expandafter\@secondoftwo
266   \else
267     \expandafter\@firstoftwo
268   \fi
269 }
270 }{
271   \def\MT@ifdimen#1{%
272     \setbox\z@=\hbox{%
273       \MT@count=1#1\relax
274       \ifnum\MT@count=\@ne
275         \aftergroup\@secondoftwo
276       \else
277         \aftergroup\@firstoftwo
278       \fi
279     }%
280 }
281 }

```

`\MT@ifdim` Test floating point numbers.

```

282 \def\MT@ifdim#1#2#3{%
283   \ifdim #1\p@ #2 #3\p@
284   \expandafter\@firstoftwo
285 \else
286   \expandafter\@secondoftwo
287 \fi
288 }

```

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

```

289 \MT@requires@pdftex5{
290   \def\MT@ifstreq#1#2{%
291     \ifcase\pdfstrcmp{#1}{#2}\relax
292     \expandafter\@firstoftwo
293   \else
294     \expandafter\@secondoftwo
295   \fi
296 }
297 }{
298   \def\MT@ifstreq#1#2{%
299     \edef\x{#1}%
300     \edef\y{#2}%
301     \ifx\x\y
302       \expandafter\@firstoftwo
303     \else
304       \expandafter\@secondoftwo

```

```

305   \fi
306   }
307 }

\MT@xadd Add item to a list.
308 \def\MT@xadd#1#2{%
309   \ifx#1\relax
310     \xdef#1{#2}%
311   \else
312     \xdef#1{#1#2}%
313   \fi
314 }

\MT@xaddb Add item to the beginning.
315 \def\MT@xaddb#1#2{%
316   \ifx#1\relax
317     \xdef#1{#2}%
318   \else
319     \xdef#1{#2#1}%
320   \fi
321 }

\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@ 322 \def\MT@map@clist@n#1#2{%
\MT@clist@function 323   \ifx\@empty#1\else
\MT@clist@break 324     \def\MT@clist@function##1{#2}%
325     \MT@map@clist@#1,\@nil,\@nnil
326   \fi
327 }
328 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
329 \def\MT@map@clist@#1,{%
330   \ifx\@nil#1%
331     \expandafter\MT@clist@break
332   \fi
333   \MT@clist@function{#1}%
334   \MT@map@clist@
335 }
336 \let\MT@clist@function\@gobble
337 \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@ 338 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break 339 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
340 \def\MT@map@tlist@#1#2{%
341   \ifx\@nnil#2\else
342     #1{#2}%
343     \expandafter\MT@map@tlist@
344     \expandafter#1%
345   \fi
346 }
347 \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 348 \newif\ifMT@inlist@
349 \def\MT@in@clist#1#2{%
350   \def\x##1,#1,##2##3\@nnil{%
351     \if##2\@empty
352       \MT@inlist@false
353     \else
354       \MT@inlist@true

```



```

355   \fi
356 }%
357 \expandafter\x\expandafter,#2,#1,\@empty\@nnil
358 }
\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!
359 \def\MT@rem@from@clist#1#2{%
360   \def\x##1,#1,##2\x{##1,##2\y}%
361   \def\y##1,\y##2\y{\ifx,##1\@empty\else##1\fi}%
362   \xdef#2{\MT@exp@two@c\x\expandafter,#2,\y,#1,\x}%
363 }
\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.
364 \def\MT@in@tlist#1#2{%
365   \MT@inlist@false
366   \def\x{#1}%
367   \MT@map@tlist@c#2\MT@in@tlist@
368 }
369 \def\MT@in@tlist@#1{%
370   \edef\y{#1}%
371   \ifx\x\y
372     \MT@inlist@true
373     \expandafter\MT@tlist@break
374   \fi
375 }
\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@@ 376 \def\MT@in@rlist#1{%
\MT@size@name 377   \MT@inlist@false
378   \MT@map@tlist@c#1\MT@in@rlist@
379 }
380 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
381 \def\MT@in@rlist@@#1#2#3{%
382   \MT@ifdim{#2}=\mone{%
383     \MT@ifdim{#1}=\MT@size
384     \MT@inlist@true
385     \relax
386   }{%
387     \MT@ifdim\MT@size<{#1}\relax{%
388       \MT@ifdim\MT@size<{#2}%
389       \MT@inlist@true
390       \relax
391     }%
392   }%
393   \ifMT@inlist@
394     \def\MT@size@name{#3}%
395     \expandafter\MT@tlist@break
396   \fi
397 }
\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 398 \def\MT@loop#1\MT@repeat{%
399   \def\MT@iterate{#1}\relax\expandafter\MT@iterate\fi}%
400   \MT@iterate \let\MT@iterate\relax
401 }
402 \let\MT@repeat\fi

```

`\MT@while@num` Execute  $\langle\#3\rangle$  from  $\langle\#1\rangle$  up to (excluding)  $\langle\#2\rangle$ .

```

403 \def\MT@while@num#1#2#3{%
404   \@tempcnta#1\relax
405   \MT@loop #3%
406   \advance\@tempcnta \@ne
407   \ifnum\@tempcnta < #2\MT@repeat
408 }

```

`\MT@do@font` Execute  $\langle\#1\rangle$  256 times.

```

409 \def\MT@do@font{\MT@while@num\z@\@cc1vi}

```

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

`\MT@increment`

```

410 \newcount\MT@count
411 \MT@requires@etex{
412   \def\MT@increment#1{\edef#1{\number\numexpr #1 + 1\relax}}
413 }{
414   \def\MT@increment#1{%
415     \MT@count=#1\relax
416     \advance\MT@count \@ne
417     \edef#1{\number\MT@count}%
418   }
419 }

```

`\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.

```

420 \MT@requires@etex{
421   \def\MT@scale#1#2#3{%
422     \ifnum #3 = \z@
423       #1=\numexpr #1 * #2\relax
424     \else
425       #1=\numexpr #1 * #2 / #3\relax
426     \fi
427   }
428 }{
429   \def\MT@scale#1#2#3{%
430     \multiply #1 #2\relax
431     \ifnum #3 = \z@ \else
432       \divide #1 #3\relax
433     \fi
434   }
435 }

```

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

```

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

```

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

`\MT@abbr@ex`

```

437 \def\MT@abbr@pr{protrusion}
438 \def\MT@abbr@ex{expansion}
439 \def\MT@abbr@pr@c{protrusion codes}
440 \def\MT@abbr@ex@c{expansion codes}
441 \def\MT@abbr@pr@inh{protrusion inheritance}
442 \def\MT@abbr@ex@inh{expansion inheritance}
443 \def\MT@abbr@n1{noligatures}

```

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

`\MT@abbr@sp@c` 444  *$\langle beta \rangle$*

`\MT@abbr@sp@inh`

`\MT@abbr@kn`

`\MT@abbr@kn@c`

`\MT@abbr@kn@inh`

```

445 \def\MT@abbr@sp{spacing}
446 \def\MT@abbr@sp@c{interword spacing codes}
447 \def\MT@abbr@sp@inh{interword spacing inheritance}
448 \def\MT@abbr@kn{kerning}
449 \def\MT@abbr@kn@c{kerning codes}
450 \def\MT@abbr@kn@inh{kerning inheritance}
451 </beta>
\MT@rba@protrusion These we also need the other way round.
\MT@rba@expansion 452 \def\MT@rba@protrusion{pr}
\MT@rba@spacing 453 \def\MT@rba@expansion{ex}
\MT@rba@kerning 454 <beta>
455 \def\MT@rba@spacing{sp}
456 \def\MT@rba@kerning{kn}
457 </beta>
\MT@features We can work on these lists to save some guards in the dtx file.
\MT@features@long 458 \def\MT@features{pr,ex%
459 <beta> ,sp,kn%
460 }
461 \def\MT@features@long{protrusion,expansion%
462 <beta> ,spacing,kerning%
463 }
\MT@is@feature Whenever an optional argument accepts a list of features, we can use this com-
mand to check whether a feature exists in order to prevent a rather confusing
'Missing \endcsname inserted' error message. The feature (long form) must be in
\@tempa, the type of list to ignore in <#1>, then comes the action.
464 \def\MT@is@feature#1{%
465 \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
466 \ifMT@inlist@
467 \expandafter\@firstofone
468 \else
469 \MT@error{\@tempa' is not an available micro-typographic\MessageBreak
470 feature. Ignoring #1}{Available features are: \MT@features@long'.}%
471 \expandafter\@gobble
472 \fi
473 }

```

### 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.

```
474 \let\MT@setup@\empty
```

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

```
475 \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).

```

476 \AtBeginDocument{\MT@setup@ \global\let\MT@setup@\empty}
\MT@with@package We almost never do anything if a package is not loaded.
477 \def\MT@with@package#1{\ifpackageloaded{#1}\@firstofone\@gobble}
\MT@pdfcprot@error Our competitor, the pdfcprot package, must not be tolerated!
478 \def\MT@pdfcprot@error{%
479 \MT@error{Detected the `pdfcprot' package!\MessageBreak
480 `~\MT@MT' and `pdfcprot' may not be used together}{%
481 The `pdfcprot' package provides an interface to character protrusion.\MessageBreak
482 So does the `~\MT@MT' package. Using both packages at the same\MessageBreak
483 time will almost certainly lead to undesired results. Have your choice!}%
484 \let\MT@pdfcprot@error\relax
485 }
486 \MT@with@package{pdfcprot}\MT@pdfcprot@error

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

Peter Wilson was so kind to provide the \l@dunhbox@line hook in ledmac to
allow for protrusion. \leftmargin kern and \rightmargin kern are new primitives
of pdfTeX 1.21b (aka. 1.30.0).
487 \MT@requires@pdftex5{
488 \def\MT@ledmac@setup{%
489 \ifMT@protrusion
490 \MT@ifdefined@c@TF\l@dunhbox@line{%
491 \MT@info@nl{Patching ledmac to enable character protrusion}%
492 \newdimen\MT@led@kern
493 \let\MT@led@unhbox@line\l@dunhbox@line
494 \renewcommand*{\l@dunhbox@line}[1]{%
495 \ifhbox##1%
496 \MT@led@kern=\rightmargin kern##1%
497 \kern\leftmargin kern##1%
498 \MT@led@unhbox@line##1%
499 \kern\MT@led@kern
500 \fi
501 }%
502 }%
503 \MT@warning@nl{%
504 Character protrusion in paragraphs with line\MessageBreak
505 numbering will only work if you update ledmac}%
506 }%
507 \fi
508 }
509 }{
510 \def\MT@ledmac@setup{%
511 \ifMT@protrusion
512 \MT@warning@nl{%
513 The pdftex version you are using does not allow\MessageBreak
514 character protrusion in paragraphs with line\MessageBreak
515 numbering by the `ledmac' package.\MessageBreak
516 Upgrade pdftex to version 1.30 or later}%
517 \fi
518 }

```

519 }

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

```
520 \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`.)

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

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

```
522 \ifpackagewith{babel}{spanish}{%
523   \MT@ifdefined@c@T\percentsign{\let\%\percentsign}%
524 } \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.)

```
525 \MT@with@package{csquotes}{%
526   \@ifpackageafter{csquotes}{2005/05/11}\@disablequotes \relax
527 }%
```

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

```
528 \ifpackageloaded{hyperref}\MT@restore@p@h{%
529   \MT@with@package{tex4ht}\MT@restore@p@h
530 }%
531 }
```

Check again at the end of the preamble.

```
532 \MT@addto@setup{%
533   \MT@with@package{pdfcpot}\MT@pdfcpot@error
534   \MT@with@package{ledmac}\MT@ledmac@setup
```

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

```
535 \let\MT@setupfont@hook\@empty
536 \ifpackagewith{babel}{spanish}{%
537   \g@addto@macro\MT@setupfont@hook{%
538     \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
539 } \relax
540 \MT@with@package{csquotes}{%
541   \@ifpackageafter{csquotes}{2005/05/11}{%
542     \g@addto@macro\MT@setupfont@hook\@disablequotes
543   }{%
544     \MT@warning@n1{%
545       Should you receive warnings about unknown slot\MessageBreak
546       numbers, try upgrading the `csquotes' package}%
547   }%
548 }%
549 \ifpackageloaded{hyperref}{%
550   \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
```

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

```

551 \pdfstringdefDisableCommands{%
552 \let\pickup@font\MT@orig@pickupfont
553 <beta>
554 \let\lsstyle\@empty
555 \def\textls#1#\@firstofone%
556 </beta>
557 }%
558 }{%
559 \MT@with@package{tex4ht}{%
560 \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
561 }%
562 }%

```

The listings package makes numbers and letters active.

```

563 \MT@with@package{listings}{%
564 \g@addto@macro\MT@begin@catcodes{%
565 \MT@while@num{"30"}{"3A"}{\catcode\@tempcnta 12\relax}%
566 \MT@while@num{"41"}{"5B"}{\catcode\@tempcnta 11\relax}%
567 \MT@while@num{"61"}{"7B"}{\catcode\@tempcnta 11\relax}%
568 }%

```

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.

```

569 \g@addto@macro\MT@setupfont@hook{%
570 \let\lst@ProcessLetter\@empty
571 }%
572 }%

```

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.

```

573 <beta>
574 \MT@with@package{soul}{%
575 \soulregister\lsstyle 0%
576 \soulregister\textls 1%
577 }%
578 </beta>
579 }

```

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

```
580 \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 \lpcode and \rpcode (protrusion) and \efcode (expansion) for each character.

```

581 \def\MT@setupfont{%
582 \ifx\MT@vinfo\MT@info@n1
583 \MT@info{Setting up font ~\MT@exp@string\MT@font'}\fi

```

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

```
584 \MT@setupfont@hook
```

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

```
585 \MT@exp@two@c\MT@split@name\string\MT@font\@n1
```

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

```
586 \MT@exp@one@n\MT@find@file\MT@family
587 \ifx\MT@familyalias\@empty \else
588 \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.)

```
589 \ifx\cf@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!

```
590 \MT@protrusion
591 \MT@expansion
```

Interword spacing and kerning.

```
592 <beta>
593 \MT@spacing
594 \MT@kerning
595 </beta>
```

Disable ligatures?

```
596 \MT@noligatures
597 }
```

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

```
\MT@encoding 598 \def\MT@split@name#1/#2/#3/#4/#5\@nil{%
\MT@family 599 \def\MT@encoding{#1}%
600 \def\MT@family{#2}%
\MT@series 601 \def\MT@series{#3}%
\MT@shape 602 \def\MT@shape{#4}%
\MT@size 603 \def\MT@size{#5}%
```

`\MT@familyalias` Alias family?

```
604 \MT@ifdefined@n@TF{MT@\MT@family @alias}%
605 {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
606 {\let\MT@familyalias\@empty}%
607 }
```

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

```
\MT@maybe@do 608 \newif\ifMT@do
609 \def\MT@maybe@do#1{%
```

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

```
610 \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).

```
611 \MT@dotrue
612 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
613 \MT@ifdefined@n@TF{MT@checklist@##1}%
614 {\csname MT@checklist@##1\endcsname}%
615 {MT@checklist@{##1}}%
616 {#1}%
617 }%
```

```

618 \else
619 \MT@dofalse
620 \fi
621 \ifMT@do

```

`\MT@feat` stores the current feature.

```

622 \def\MT@feat{#1}%
623 \csname MT@set@#1@codes\endcsname
624 \else
625 \MT@vinfo{... No \nameuse{MT@abbr@#1}}%
626 \fi
627 }

```

`\MT@checklist@` The generic test.

```

628 \def\MT@checklist@#1#2{%
629 \edef\@tempa{\csname MT@#2@setname\endcsname}%
630 !debug \MT@ifdefined@n@T
631 debug \MT@ifdefined@n@TF
632 {\MT@#2list@#1@\@tempa}%

```

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

```

633 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
634 \csname MT@#1\expandafter\endcsname
635 \csname MT@#2list@#1@\@tempa\endcsname
636 \ifMT@inlist@
637 debug \MT@dinfo@n1{1}{\nameuse{MT@abbr@#2}: #1 `\debug \MT@dinfo@n1{1}{\nameuse{MT@abbr@#2}: #1 `\

```

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.

```

645 debug {\MT@dinfo@n1{1}{\nameuse{MT@abbr@#2}: #1 list empty}}%
646 }

```

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

```

647 \def\MT@checklist@font#1{%
648 \edef\@tempa{\csname MT@#1@setname\endcsname}%
649 !debug \MT@ifdefined@n@T
650 debug \MT@ifdefined@n@TF
651 {\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.

```

652 \expandafter\expandafter\expandafter\MT@exp@one@n
653 \expandafter\expandafter\expandafter\MT@in@clist
654 \expandafter\expandafter\expandafter{%
655 \expandafter\expandafter\expandafter\string
656 \expandafter\MT@font\expandafter}%
657 \csname MT@#1list@font@\@tempa\endcsname
658 \ifMT@inlist@
659 debug \MT@dinfo@n1{1}{\nameuse{MT@abbr@#1}: font `\debug \MT@dinfo@n1{1}{\nameuse{MT@abbr@#1}: font `\

```



```

664 \fi
665 }%
666 <debug> {\MT@dinfo@n1}{\@nameuse{MT@abbr#1}: font list empty}}%
667 }

```

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

```

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

```

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

```

697 \def\MT@checklist@size#1{%
698 \edef\@tempa{\csname MT@#1@setname\endcsname}%
699 <!debug> \MT@ifdefined@nT
700 <debug> \MT@ifdefined@nTF
701 {MT@#1list@size@\@tempa}{%
702 \expandafter\MT@in@rlist
703 \csname MT@#1list@size@\@tempa\endcsname
704 \ifMT@inlist@
705 <debug>\MT@dinfo@n1}{\@nameuse{MT@abbr#1}: size ` \MT@size' in list}%
706 \MT@dotrue
707 \else
708 <debug>\MT@dinfo@n1}{\@nameuse{MT@abbr#1}: size ` \MT@size' not in list}%
709 \MT@dofalse
710 \expandafter\MT@clist@break
711 \fi
712 }%
713 <debug> {\MT@dinfo@n1}{\@nameuse{MT@abbr#1}: size list empty}}%
714 }

```

### 13.4.1 Protrusion

`\MT@protrusion` Set up for protrusion?

```

715 \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.

```
716 \def\MT@set@pr@codes{%
    Check whether and if, which list should be applied to the current font.
717 \MT@if@list@exists{%
718 \MT@get@dimen@six
719 \MT@get@opt
720 \MT@reset@pr@codes

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

    Load additional lists?
722 \MT@load@list\MT@pr@c@name
723 \MT@set@listname

    Load the main list.
724 \MT@let@cn\@tempc{MT@pr@c@\MT@pr@c@name}%
725 \expandafter\MT@pr@do\@tempc,\relax,%
726 }\MT@reset@pr@codes
727 }
```

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

```
728 \def\MT@set@all@pr#1#2{%
729 <debug>\MT@info@nl{3}{-- lp/rp: setting all to \number#1/\number#2}%
730 \MT@do@font{%
731 \lpcode\MT@font\@tempcnta=#1\relax
732 \rpcode\MT@font\@tempcnta=#2\relax
733 }%
734 }
```

`\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.

```
735 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
736 \let\MT@reset@pr@codes\relax
```

`\MT@gobble@settings` If `\fontdimen 6` 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
737 \def\MT@get@dimen@six{%
738 \ifnum\fontdimen6\MT@font=\z@
739 \MT@warning@nl{%
740 Font '\MT@exp@string\MT@font' does not specify its\MessageBreak
741 \backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
742 \nameuse{MT@abbr@\MT@feat} will not work with this font}%
743 \expandafter\MT@gobble@settings
744 \else
745 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
746 \fi
747 }
748 \def\MT@gobble@settings#1\@tempc,\relax,{}

\MT@pr@do Split up the values and set \lpcode and \rptide.
749 \def\MT@pr@do#1,{%
750 \ifx\relax#1\@empty\else
751 \MT@pr@split #1=\relax
752 \expandafter\MT@pr@do
753 \fi
```

```
754 }
```

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

```
755 \def\MT@pr@split#1=#2=#3\relax{%
756   \def\@tempa{#1}%
757   \ifx\@tempa@empty \else
758     \MT@get@slot
759     \ifnum\MT@char > \m@ne
760       \MT@get@char@unit
761       \MT@pr@split@val#2\relax
762     \fi
763   \fi
764 }
```

`\MT@pr@split@val`

```
765 \def\MT@pr@split@val#1,#2\relax{%
766   \def\@tempb{#1}%
767   \MT@ifempty\@tempb\relax{%
768     \MT@scale@to@em
769     \lpcode\MT@font\MT@char=\@tempcntb
770   }%
771   \def\@tempb{#2}%
772   \MT@ifempty\@tempb\relax{%
773     \MT@scale@to@em
774     \rptide\MT@font\MT@char=\@tempcntb
775   }%
776   \debug\MT@dinfo@n1{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char: [#1]}%
777 }
```

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

```
778 \MT@ifdefined@cT\MT@pr@inh@name{%
779   \MT@ifdefined@nT{MT@inh@MT@pr@inh@name @\MT@char @}{%
780     \expandafter\MT@map@tlist@c
781     \csname MT@inh@MT@pr@inh@name @\MT@char @\endcsname
782     \MT@set@pr@heirs
783   }%
784 }%
785 }
```

`\MT@scale@to@em`

Since pdfTeX version 0.14h, we have to adjust the protrusion factors (i. e., convert numbers from thousandths of character width to thousandths of an *em* of the font). We have to do this *before* setting the inheriting characters, so that the latter inherit the absolute value, not the relative one if they have a differing width (e. g., the ‘ff’ ligature). Unlike `protcode.tex` and `pdfcprot`, we do not calculate with `\lpcode` resp. `\rptide`, since this would disallow protrusion factors larger than the character width (since `\l[1r]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`.

```
786 \MT@requires@pdfTeX3{
787   \def\MT@scale@to@em{%
788     \@tempcntb=\MT@count\relax
```

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

```

789 \MT@scale\@tempcntb \@tempb \MT@dimen@six
790 \ifnum\@tempcntb=\z@ \else
791 \MT@scale@factor
792 \fi
793 }

```

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

```

794 \MT@requires@etex{
795 \def\MT@get@charwd{%
796 \MT@count=\fontcharwd\MT@font\MT@char\relax
797 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
798 }
799 }{
800 \def\MT@get@charwd{%
801 \setbox\z@=\hbox{\MT@font \char\MT@char}%
802 \MT@count=\wd\z@\relax
803 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
804 }
805 }

```

No adjustment with versions 0.14f and 0.14g.

```

806 }{
807 \def\MT@scale@to@em{%
808 \MT@count=\@tempb\relax
809 \ifnum\MT@count=\z@ \else
810 \MT@scale@factor
811 \fi
812 }

```

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

```

813 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
814 }

```

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

```

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

```

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

```

824 \MT@requires@etex{
825 \def\MT@info@missing@char{%
826 \MT@info@n{Character `the\MT@toks' \iffontchar\MT@font\MT@char
827 has a width of Opt \else is missing \fi in font\MessageBreak
828 \MT@exp@string\MT@font'. Ignoring protrusion settings\MessageBreak
829 for this character}%
830 }
831 }{
832 \def\MT@info@missing@char{%
833 \MT@info@n{%
834 Character `the\MT@toks' has a width of Opt\MessageBreak
835 (it's probably missing) in font \MT@exp@string\MT@font'.\MessageBreak
836 Ignoring protrusion settings for this character}%
837 }
838 }

```

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

```
839 \def\MT@scale@factor{%
840   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
841     \expandafter\MT@scale\expandafter
842       \@tempcntb \csname MT@\MT@feat @factor@\endcsname \@m
843   \fi
844   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
845     \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @max\endcsname
846   \else
847     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
848       \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @min\endcsname
849     \fi
850   \fi
851 }
```

`\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.

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

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

```
865 \def\MT@get@opt{%
866   \MT@set@listname
```

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

```
\MT@sp@factor@ 867 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 868   \MT@let@nn{MT@\MT@feat @factor@}
869     {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
870   \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
871     \number\csname MT@\MT@feat @factor@\endcsname/1000}%
872   }{%
873     \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
874   }%
```

`\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@ 875 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
876   \MT@let@nn{MT@\MT@feat @unit@}
877     {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
878   \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\@empty
879     \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
880       relative to character widths}%
881   \else
882     \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\@m@ne
883     \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
884       relative to width of space}%
885   \fi
886 \fi
```

```

887 }{%
888 \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
889 }%

```

`\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`

```

890 \let\MT@get@char@unit\relax
891 \let\MT@get@space@unit\@gobble
892 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\@empty
893 \let\MT@get@char@unit\MT@get@charwd
894 \else
895 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\m@ne
896 \let\MT@get@space@unit\MT@get@font@dimen
897 \else
898 \expandafter\MT@get@unit\csname MT@\MT@feat @unit@\endcsname
899 \fi
900 \fi

```

We might have to change the input encoding,

`\MT@cat` for which we need the type of list we're currently loading.

```

901 \def\MT@cat{c}%
902 \MT@set@inputenc

```

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

```

903 \MT@ifdefined@nT{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
904 \csname MT@preset@\MT@feat\endcsname
905 \MT@let@nc{MT@reset@\MT@feat @codes}\relax
906 }%
907 }

```

`\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.

`\MT@get@unit@`

```

908 \def\MT@get@unit#1{%
909 \expandafter\MT@get@unit@#1 e!\@nil
910 \ifx\x\@empty\else\let#1\x\fi
911 \@defaultunits\@tempdima#1 pt\relax\@nnil
912 \ifdim\@tempdima=\z@
913 \MT@warning@n1{%
914 Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
915 width. Setting factors of list ~\@nameuse{MT@\MT@feat @c@name}'\MessageBreak
916 relative to character widths instead}%
917 \let#1\@empty
918 \let\MT@get@char@unit\MT@get@charwd
919 \else
920 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
921 to \the\@tempdima}%
922 \MT@count=\@tempdima\relax
923 \fi
924 }
925 \def\MT@get@unit@#1e#2#3\@nil{%
926 \ifx\#3\\\let\x\@empty \else
927 \if m#2%
928 \edef\x{#1\fontdimen6\MT@font}%
929 \else
930 \if x#2%
931 \edef\x{#1\fontdimen5\MT@font}%
932 \fi

```

```

933 \fi
934 \fi
935 }

```

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

```

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

```

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

```

951 \def\MT@set@pr@heirs#1{%
952 \lpcode\MT@font#1=\lpcode\MT@font\MT@char
953 \rprcode\MT@font#1=\rprcode\MT@font\MT@char
954 <*debug>
955 \MT@info@n1{2}{-- heir of \MT@char: #1}%
956 \MT@info@n1{4}{;;; lp/rp (#1): \number\lpcode\MT@font\MT@char/%
957 \number\rprcode\MT@font\MT@char}%
958 </debug>
959 }

```

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

```

\MT@preset@pr@
960 \def\MT@preset@pr{%
961 \expandafter\expandafter\expandafter\MT@preset@pr@
962 \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
963 }
964 \def\MT@preset@pr@#1,#2\@nil{%
965 \ifx\MT@pr@unit@\@empty
966 \MT@warn@preset@t@width{pr}%
967 \let\MT@preset@aux\MT@preset@aux@factor
968 \else
969 \let\MT@preset@aux\MT@preset@aux@space
970 \fi
971 \MT@preset@aux{#1}\@tempa
972 \MT@preset@aux{#2}\@tempb
973 \MT@set@all@pr\@tempa\@tempb
974 }

```

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

```

\MT@preset@aux@factor
\MT@preset@aux@space
975 \def\MT@preset@aux@factor#1#2{%
976 \@tempcntb=#1\relax
977 \MT@scale@factor
978 \edef#2{\number\@tempcntb}%
979 }
980 \def\MT@preset@aux@space#1#2{%
981 \def\@tempb{#1}%
982 \MT@get@space@unit\tw@
983 \MT@scale@to@em
984 \edef#2{\number\@tempcntb}%
985 }

```

`\MT@warn@preset@tewidth`

```
986 \def\MT@warn@preset@tewidth#1{%
987   \MT@warning@n1{%
988     Cannot preset characters relative to their widths\MessageBreak
989     for \nameuse{MT@abbr@#1} list `\

```

### 13.4.2 Expansion

`\MT@expansion` Set up for expansion?

```
992 \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).

```
993 \def\MT@set@ex@codes@s{%
994   \MT@if@list@exists{%
995     \MT@get@ex@opt
996     \MT@reset@ef@codes
997     \MT@get@inh@list
998     \MT@load@list\MT@ex@cc@name
999     \MT@set@listname
1000    \MT@let@cn\@tempc{MT@ex@cc\MT@ex@cc@name}%
1001    \expandafter\MT@ex@do\@tempc,\relax,%
1002    \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1003   }\relax
1004 }
```

`\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.

```
1005 \newif\ifMT@nonselected
1006 \def\MT@set@ex@codes@n{%
1007   \MT@nonselectedtrue
1008   \MT@if@list@exists
1009   \MT@get@ex@opt
1010   {%
1011     \let\MT@stretch@\MT@stretch
1012     \let\MT@shrink@\MT@shrink
1013     \let\MT@step@\MT@step
1014     \let\MT@auto@\MT@auto
1015     \let\MT@ex@factor@\MT@ex@factor
1016   }%
1017   \MT@reset@ef@codes
1018   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1019   \MT@nonselectedfalse
1020 }
```

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

```
1021 \let\MT@set@ex@codes\MT@set@ex@codes@n
```

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

`\MT@reset@ef@codes@`

```
1022 \def\MT@set@all@ex#1{%
1023   <debug>\MT@info@n1{3}{-- ex: setting all to \number#1}%
```



```

1024 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1025 }
1026 \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.

```

1027 \MT@requires@pdfTeX4{
1028   \def\MT@reset@ef@codes{%
1029     \ifnum\MT@ex@factor@=\@m \else
1030       \MT@reset@ef@codes@
1031     \fi
1032   }
1033 }{
1034   \let\MT@reset@ef@codes\MT@reset@ef@codes@
1035 }

```

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

```

1036 \def\MT@ex@do#1,{%
1037   \ifx\relax#1\@empty \else
1038     \MT@ex@split #1=\relax
1039     \expandafter\MT@ex@do
1040   \fi
1041 }

```

\MT@ex@split

```

1042 \def\MT@ex@split#1=#2=#3\relax{%
1043   \def\@tempa{#1}%
1044   \ifx\@tempa\@empty \else
1045     \MT@get@slot
1046     \ifnum\MT@char > \m@ne
1047       \@tempcntb=#2\relax

```

Take an optional factor into account.

```

1048     \ifnum\MT@ex@factor@=\@m \else
1049       \MT@scale\@tempcntb \MT@ex@factor@ \@m
1050     \fi
1051     \ifnum\@tempcntb > \MT@ex@max
1052       \MT@warn@ex@too@large\MT@ex@max
1053     \else
1054       \ifnum\@tempcntb < \MT@ex@min
1055         \MT@warn@ex@too@large\MT@ex@min
1056       \fi
1057     \fi
1058     \efcode\MT@font\MT@char=\@tempcntb
1059 (debug) \MT@edinfo{n{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#2]}%

```

Heirs, heirs, I love thy heirs.

```

1060     \MT@ifdefined@c@T\MT@ex@inh@name{%
1061       \MT@ifdefined@n@T\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1062         \expandafter\MT@map@tlist@c
1063         \csname MT@inh@\MT@ex@inh@name @\MT@char @\endcsname
1064         \MT@set@ex@heirs
1065       }%
1066     }%
1067   \fi
1068 \fi
1069 }

```

\MT@warn@ex@too@large

```

1070 \def\MT@warn@ex@too@large#1{%
1071   \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1072     character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1073     Setting it to the maximum of \number#1}%

```

```

1074 \@tempcntb=#1\relax
1075 }

\MT@get@ex@opt Apply different values to this font?
\MT@ex@factor@ 1076 \def\MT@get@ex@opt{%
\MT@stretch@ 1077 \MT@set@listname
1078 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1079 \MT@let@cn\MT@ex@factor@{MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1080 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 1081 }{%
1082 \let\MT@ex@factor@\MT@ex@factor
1083 }%
1084 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1085 \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1086 \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
1087 \def\@tempa{autoexpand}%
1088 \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1089 \def\MT@cat{c}%
1090 \MT@set@inputenc
1091 \MT@ifdefined@n@T{MT@ex@c@\csname MT@ex@c@name\endcsname @preset}{%
1092 \MT@preset@ex
1093 \let\MT@reset@ef@codes\relax
1094 }%
1095 }

\MT@get@ex@opt@
1096 \def\MT@get@ex@opt@#1#2{%
1097 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}{%
1098 \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
1099 \MT@vinfo{... : #2}%
1100 }{%
1101 \MT@let@nn{MT@#1@}{MT@#1}%
1102 }%
1103 }

\MT@set@ex@heirs
1104 \def\MT@set@ex@heirs#1{%
1105 \efcode\MT@font#1=\efcode\MT@font\MT@char
1106 *debug
1107 \MT@dinfo@n1{2}{-- heir of \MT@char: #1}%
1108 \MT@dinfo@n1{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1109 /debug
1110 }

\MT@preset@ex
1111 \def\MT@preset@ex{%
1112 \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
1113 \MT@scale@factor
1114 \MT@set@all@ex@\@tempcntb
1115 }
1116 *beta

```

### 13.4.3 Interword Space (Glue)

```

\MT@spacing Adjustment of interword spacing? Only for sufficiently new versions of pdfTeX.
1117 \MT@requires@pdf@tex6{
1118 \def\MT@spacing{\MT@maybe@do{sp}}
1119 }{
1120 \let\MT@spacing\relax
1121 }

```

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

```

1122 \def\MT@set@sp@codes{%
1123   \MT@if@list@exists{%
1124     \MT@get@dimen@six
1125     \MT@get@opt
1126     \MT@reset@sp@codes
1127     \MT@get@inh@list
1128     \MT@load@list\MT@sp@c@name
1129     \MT@set@listname
1130     \MT@let@cn\@tempc{\MT@sp@c@\MT@sp@c@name}%
1131     \expandafter\MT@sp@do\@tempc,\relax,%
1132   }\MT@reset@sp@codes
1133 }

```

`\MT@sp@do`

```

1134 \def\MT@sp@do#1,{%
1135   \ifx\relax#1\@empty \else
1136     \MT@sp@split #1=\relax
1137     \expandafter\MT@sp@do
1138   \fi
1139 }

```

`\MT@sp@split`

```

1140 \def\MT@sp@split#1=#2=#3\relax{%
1141   \def\@tempa{#1}%
1142   \ifx\@tempa\@empty \else
1143     \MT@get@slot
1144     \ifnum\MT@char > \m@ne
1145       \MT@get@char@unit
1146       \MT@sp@split@val#2\relax
1147     \fi
1148   \fi
1149 }

```

`\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).

```

1150 \def\MT@sp@split@val#1,#2,#3\relax{%
1151   \def\@tempb{#1}%
1152   \MT@ifempty\@tempb\relax{%
1153     \MT@get@space@unit\tw@
1154     \MT@scale@to@em
1155     \kerncode\MT@font\MT@char=\@tempcntb
1156   }(debug)\MT@info@nl{4}{;;; knbs (\MT@char): \number\kerncode\MT@font\MT@char: [#1]}%
1157   }%
1158   \def\@tempb{#2}%
1159   \MT@ifempty\@tempb\relax{%
1160     \MT@get@space@unit\thr@
1161     \MT@scale@to@em
1162     \stbscode\MT@font\MT@char=\@tempcntb
1163   }(debug)\MT@info@nl{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1164   }%
1165   \def\@tempb{#3}%
1166   \MT@ifempty\@tempb\relax{%
1167     \MT@get@space@unit4%
1168     \MT@scale@to@em
1169     \shbscode\MT@font\MT@char=\@tempcntb
1170   }(debug)\MT@info@nl{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1171   }%
1172   \MT@ifdefined@c@T\MT@sp@inh@name{%
1173     \MT@ifdefined@n@T\MT@inh@\MT@sp@inh@name @\MT@char @}%
1174     \expandafter\MT@map@tlist@c
1175     \csname MT@inh@\MT@sp@inh@name @\MT@char @\endcsname

```

```

1176     \MT@set@sp@heirs
1177     }%
1178 }%
1179 }

\MT@set@sp@heirs

1180 \def\MT@set@sp@heirs#1{%
1181   \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1182   \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1183   \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1184   < *debug >
1185   \MT@info@n1{2}{-- heir of \MT@char: #1}%
1186   \MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1187                                     \number\stbscode\MT@font\MT@char/%
1188                                     \number\shbscode\MT@font\MT@char}%
1189   < /debug >
1190 }

\MT@set@all@sp

\MT@reset@sp@codes
\MT@reset@sp@codes@ 1191 \def\MT@set@all@sp#1#2#3{%
1192   < debug > \MT@info@n1{3}{-- knbs/stbs/shbs: setting all to \number#1/\number#2/\number#3}%
1193   \MT@do@font{%
1194     \knbscode\MT@font\@tempcnta=#1\relax
1195     \stbscode\MT@font\@tempcnta=#2\relax
1196     \shbscode\MT@font\@tempcnta=#3\relax
1197   }%
1198 }
1199 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1200 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1201 \def\MT@preset@sp{%
1202   \expandafter\expandafter\expandafter\MT@preset@sp@
1203   \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1204 }
1205 \def\MT@preset@sp@#1,#2,#3\@nil{%
1206   \ifx\MT@sp@unit@\empty
1207     \MT@warn@preset@t@width{sp}%
1208     \MT@preset@aux@factor{#1}\@tempa
1209     \MT@preset@aux@factor{#2}\@tempc
1210     \MT@preset@aux@factor{#3}\@tempb
1211   \else
1212     \MT@preset@aux@space{#1}\@tempa
1213     \def\@tempb{#2}%
1214     \MT@get@space@unit\thr@
1215     \MT@scale@to@em
1216     \edef\@tempc{\number\@tempcntb}%
1217     \def\@tempb{#3}%
1218     \MT@get@space@unit4%
1219     \MT@scale@to@em
1220     \edef\@tempb{\number\@tempcntb}%
1221   \fi
1222   \MT@set@all@sp\@tempa\@tempc\@tempb
1223 }

```

### 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.

```

1224 \MT@requires@pdf@tex6{
1225   \def\MT@kerning{\MT@maybe@do{kn}}
1226 }{

```

```

1227 \let\MT@kerning\relax
1228 }

\MT@set@kn@codes It's getting boring, I know.
1229 \def\MT@set@kn@codes{%
1230 \MT@if@list@exists{%
1231 \MT@get@dimen@six
1232 \MT@get@opt
1233 \MT@reset@kn@codes
1234 \MT@get@inh@list
1235 \MT@load@list\MT@kn@cc@name
1236 \MT@set@listname
1237 \MT@let@cn\@tempc{MT@kn@cc\MT@kn@cc@name}%
1238 \expandafter\MT@kn@do\@tempc,\relax,%
1239 }\MT@reset@kn@codes
1240 }

\MT@kn@do
1241 \def\MT@kn@do#1,{%
1242 \ifx\relax#1\@empty \else
1243 \MT@kn@split #1=\relax
1244 \expandafter\MT@kn@do
1245 \fi
1246 }

\MT@kn@split
1247 \def\MT@kn@split#1=#2=#3\relax{%
1248 \def\@tempa{#1}%
1249 \ifx\@tempa\@empty \else
1250 \MT@get@slot
1251 \ifnum\MT@char > \m@ne
1252 \MT@get@char@unit
1253 \MT@kn@split@val#2\relax
1254 \fi
1255 \fi
1256 }

\MT@kn@split@val Again, the unit may be measured in the space dimension; this time only \fontdimen 2.
1257 \def\MT@kn@split@val#1,#2\relax{%
1258 \def\@tempb{#1}%
1259 \MT@ifempty\@tempb\relax{%
1260 \MT@get@space@unit\tw@
1261 \MT@scale@to@em
1262 \knbccode\MT@font\MT@char=\@tempcntb
1263 <debug>\MT@dinfo@n1{4}{;;; knbc (\MT@char): \number\knbccode\MT@font\MT@char: [#1]}%
1264 }%
1265 \def\@tempb{#2}%
1266 \MT@ifempty\@tempb\relax{%
1267 \MT@get@space@unit\tw@
1268 \MT@scale@to@em
1269 \knaccode\MT@font\MT@char=\@tempcntb
1270 <debug>\MT@dinfo@n1{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1271 }%
1272 \MT@ifdefined@c@T\MT@kn@inh@name{%
1273 \MT@ifdefined@n@T{MT@inh@MT@kn@inh@name @\MT@char @}{%
1274 \expandafter\MT@map@tlist@c
1275 \csname MT@inh@MT@kn@inh@name @\MT@char @\endcsname
1276 \MT@set@kn@heirs
1277 }%
1278 }%
1279 }

```

`\MT@set@kn@heirs`

```
1280 \def\MT@set@kn@heirs#1{%
1281   \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1282   \knaccode\MT@font#1=\knaccode\MT@font\MT@char
1283   *debug
1284   \MT@info{n1}{2}{-- heir of \MT@char: #1}%
1285   \MT@info{n1}{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1286                               \number\knaccode\MT@font\MT@char}%
1287 /debug
1288 }
```

`\MT@set@all@kn`

```
\MT@reset@kn@codes 1289 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1290 debug\MT@info{n1}{3}{-- knac/knbc: setting all to \number#1/\number#2}%
1291   \MT@do@font{%
1292     \knbcode\MT@font\@tempcnta=#1\relax
1293     \knaccode\MT@font\@tempcnta=#2\relax
1294   }%
1295 }
1296 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1297 \let\MT@reset@kn@codes\relax
```

`\MT@preset@kn`

```
\MT@preset@kn@ 1298 \def\MT@preset@kn{%
1299   \expandafter\expandafter\expandafter\MT@preset@kn@
1300   \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1301 }
1302 \def\MT@preset@kn@#1,#2\@nil{%
1303   \ifx\MT@kn@unit@\empty
1304     \MT@warn@preset@tewidth{kn}%
1305     \let\MT@preset@aux\MT@preset@aux@factor
1306   \else
1307     \let\MT@preset@aux\MT@preset@aux@space
1308   \fi
```

#### Letterspacing factor.

```
1309 \MT@ifstreq\MT@kn@context{letterspacing}{%
1310   \@tempcnta\MT@kn@factor@\relax
1311   \MT@scale@\tempcnta \MT@letterspacing@ \@m
1312   \edef\MT@kn@factor@{\number\@tempcnta}%
1313 } \relax
1314 \MT@preset@aux{#1}\@tempa
1315 \MT@preset@aux{#2}\@tempb
1316 \MT@set@all@kn\@tempa\@tempb
1317 }
```

### 13.4.5 Letterspacing

`\lssstyle` Letterspacing is a special case of extra kerning. It will temporarily switch kerning on, activate the `all` font set and load the `letterspacing` context.

```
1318 \MT@requires@pdftex6{
1319   \DeclareRobustCommand\lssstyle{%
1320     \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 `\lssstyle`.

```
1321     \begingroup
1322     \escapechar@m@ne
1323     \expandafter\MT@exp@two@n\expandafter\MT@in@clist\expandafter
1324     {\expandafter\string\expandafter
```

```

1325         \csname\curr@fontshape/\f@size\expandafter\endcsname}%
1326         {\csname MT@knlst@font@MT@kn@setname\endcsname}%
1327     \ifMT@inlist@ \else
1328         \expandafter\MT@xadd
1329         \csname MT@knlst@font@MT@kn@setname\endcsname
1330         {\MT@exp@string\csname\curr@fontshape/\f@size\endcsname,}%
1331     \fi
1332 \endgroup
1333 \fi
1334 \MT@kerningtrue
1335 \pdfappendkern\@ne
1336 \pdfprependkern\@ne
1337 \def\MT@kn@setname{all}%
1338 \MT@ifdefined@c@TF\MT@letterspacing@ \relax{%
1339     \let\MT@letterspacing@\MT@letterspacing
1340 }%
1341 \microtypecontext{kerning=letterspacing}%
1342 }
1343 }{
1344 \DeclareRobustCommand\lsstyle{%
1345     \MT@warning{Letterspacing only works with pdftex version 1.40\MessageBreak
1346         or newer. You might want to use the `soul' package\MessageBreak
1347         instead}%
1348     \global\let\lsstyle\relax
1349 }
1350 }

```

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

```

1351 \DeclareRobustCommand\textls[2][ ]{%
1352     \MT@ifempty{#1}%
1353     {\let\MT@letterspacing@\undefined}%
1354     {\KV@esp@def\MT@letterspacing@{#1 }}%
1355     {\lsstyle #2}%
1356 }
1357 /beta

```

### 13.4.6 Disabling Ligatures

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

```

1358 \MT@requires@pdftex5{
1359     \def\MT@noligatures{%
1360         \ifMT@noligatures
1361             \MT@dotrue
1362             \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1363                 \MT@ifdefined@n@TF{MT@checklist@##1}%
1364                 {\csname MT@checklist@##1\endcsname}%
1365                 {\MT@checklist@{##1}}%
1366             }{n1}%
1367         }%
1368     \else
1369         \MT@dofalse
1370     \fi
1371     \ifMT@do
1372         \pdfnoligatures\MT@font
1373         \MT@vinfo{... Disabling ligatures}%
1374     \fi
1375 }
1376 }{
1377     \let\MT@noligatures\relax
1378 }

```

### 13.4.7 Loading the Configuration

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

```

1379 \def\MT@load@list#1{%
1380   \edef\@tempa{#1}%
1381   \MT@let@cn\@tempb{MT@\MT@feat @c@\@tempa load}%
1382   \MT@ifstreq\@tempa\@tempb{%
1383     \MT@warning{\@nameuse{MT@abbr@\MT@feat} list `@\@tempa' cannot load itself}%
1384   }{%
1385     \ifx\@tempb\relax \else
1386       \MT@ifdefined@n@TF{MT@\MT@feat @c@\@tempb}{%
1387         \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list `@\@tempb'}%
1388         \begingroup
1389           \MT@load@list\@tempb
1390         \endgroup
1391         \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\MessageBreak`@\@tempb'}%
1392         \MT@let@cn\@tempc{MT@\MT@feat @c@\@tempb}%
1393         \expandafter\csname MT@\MT@feat @do\expandafter\endcsname\@tempc,\relax,%
1394       }{%
1395         \MT@warning{\@nameuse{MT@abbr@\MT@feat} list `@\@tempb' undefined.
1396           Cannot load\MessageBreak it from list `@\@tempa'}%
1397       }%
1398     \fi
1399   }%
1400 }
```

`\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.

```

1401 \let\MT@file@list\@empty
1402 \def\MT@find@file#1{%
```

Check for existence of the file only once.

```

1403   \MT@in@clist{#1}\MT@file@list
1404   \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.

```

1405   \MT@begin@catcodes
1406     \let\MT@begin@catcodes\relax
1407     \let\MT@end@catcodes\relax
1408     \InputIfFileExists{mt-#1.cfg}{%
1409       \edef\MT@curr@file{mt-#1.cfg}%
1410       \MT@vinfo{... Loading configuration file \MT@curr@file}%
1411       \MT@xadd\MT@file@list{#1,%
1412     }{%
1413       \expandafter\MT@get@basefamily#1\relax\relax\relax
1414       \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1415       \ifMT@inlist\else
1416         \InputIfFileExists{mt-\@tempa.cfg}{%
1417           \edef\MT@curr@file{mt-\@tempa.cfg}%
1418           \MT@vinfo{... Loading configuration file \MT@curr@file}%
1419           \MT@xadd\MT@file@list{\@tempa,#1,%
1420         }{%
1421           \MT@vinfo{... No configuration file mt-#1.cfg}%
1422           \MT@xadd\MT@file@list{#1,%
1423         }%
1424       \fi
1425     }%
1426   \endgroup
1427 \fi
1428 }
```



`\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  $\LaTeX$  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.

```

1429 \def\MT@begin@catcodes{%
1430   \begingroup
1431   \makeatletter
1432   \catcode\^7%
1433   \catcode\ 9%
1434   \catcode\^^I9%
1435   \catcode\^^M9%
1436   \catcode\\\z@
1437   \catcode\{\@ne
1438   \catcode\}\@tw@
1439   \catcode\#6%
1440   \catcode\%14%
1441   \MT@map@tlist@n
1442   {\!\!"\$\&\'\(\)\*\+!,\-\.\./\:\;\<=\>?\[\]\_-\`\/\~}%
1443   \@makeother

```

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

```

1444 \let\KV@sp@def\def
1445 }

```

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

```

1446 \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...`).

```

1447 \def\MT@get@basefamily#1#2#3#4\relax{%
1448   \ifx#2\relax \def\@tempa{#1}\else
1449     \ifx#3\relax \def\@tempa{#1#2}\else
1450       \def\@tempa{#1#2#3}%
1451       \ifx\relax#4\relax \else
1452         \MT@ifstreq{#4}{\string x}\relax{%
1453           \MT@ifstreq{#4}{\string j}\relax{%
1454             \MT@ifstreq{#4}{\string w}\relax{%
1455               \def\@tempa{#1#2#3#4}}}\fi\fi\fi
1456 }

```

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

`\MT@get@listname` font.

```

\MT@get@listname@
1457 \def\MT@get@listname#1{%
1458   debug\MT@dinfo@n1{1}{trying to find \@nameuse{MT@abbr@#1} list for font \MT@font}%
1459   \let\MT@listname\undefined
1460   \def\@tempb{#1}%
1461   \MT@map@tlist@c\MT@try@order\MT@get@listname@
1462 }
1463 \def\MT@get@listname@#1{%
1464   \expandafter\MT@next@listname#1%
1465   \ifx\MT@listname\undefined \else

```

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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```

1466 \expandafter\MT@tlist@break
1467 \fi
1468 }

```

\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.

```

1469 \def\MT@try@order{%
1470 {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
1471 {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
1472 }

```

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

```

1473 \def\MT@next@listname#1#2#3#4{%
1474 \edef\@tempa{\MT@encoding
1475 \ifnum#1=\@ne \MT@family\fi
1476 \ifnum#2=\@ne \MT@series\fi
1477 \ifnum#3=\@ne \MT@shape\fi
1478 \ifnum#4=\@ne *\fi
1479 \MT@context}%
1480 (debug)\MT@info@n1{1}{trying \@tempa}%
1481 \MT@ifdefined@nTF{\MT@\@tempb @\@tempa}{%
1482 \MT@next@listname@#4%
1483 }{%

```

Also try with an alias family.

```

1484 \ifnum#1=\@ne
1485 \ifx\MT@familyalias\@empty \else
1486 \edef\@tempa{\MT@encoding
1487 \MT@familyalias
1488 \ifnum#2=\@ne \MT@series\fi
1489 \ifnum#3=\@ne \MT@shape\fi
1490 \ifnum#4=\@ne *\fi
1491 \MT@context}%
1492 (debug)\MT@info@n1{1}{(alias) \@tempa}%
1493 \MT@ifdefined@nTF{\MT@\@tempb @\@tempa}{%
1494 \MT@next@listname@#4%
1495 }%
1496 \fi
1497 \fi
1498 }%
1499 }

```

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

```

1500 \def\MT@next@listname@#1{%
1501 \ifnum#1=\@ne
1502 \expandafter\MT@in@rlist\csize MT@\@tempb @\@tempa @size@endcsize
1503 \ifMT@inlist@
1504 \let\MT@listname\MT@size@name
1505 \fi

```

```

1506 \else
1507 \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
1508 \fi
1509 }

```

\MT@if@list@exists

```

\MT@context 1510 \def\MT@if@list@exists{%
1511 \MT@let@cn\MT@context{MT@\MT@feat @context}%
1512 \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
1513 \MT@get@listname{\MT@feat @c}%
1514 \MT@ifdefined@c@TF\MT@listname{%
1515 \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
1516 \ifMT@nonselected
1517 \MT@vinfo{... Applying non-selected expansion (list ~\MT@ex@c@name')}%
1518 \else
1519 \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list
1520 \~\@nameuse{MT@\MT@feat @c@name}'}%
1521 \fi
1522 \@firstoftwo
1523 }%

```

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

```

1524 \MT@let@nc{MT@\MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

1525 \ifMT@nonselected
1526 \MT@vinfo{... Applying non-selected expansion}%
1527 \else
1528 \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
1529 for font\MessageBreak~\MT@exp@string\MT@font%
1530 \ifx\MT@context\@empty\else\space(context: ~\MT@context')\fi.
1531 Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
1532 \fi
1533 \@secondoftwo
1534 }%
1535 }

```

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

```

\MT@context 1536 \def\MT@get@inh@list{%
1537 \let\MT@context\@empty
1538 \MT@get@listname{\MT@feat @inh}%
1539 \MT@ifdefined@c@TF\MT@listname{%
1540 \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
1541 <debug>
1542 \MT@dinfo@n1{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
1543 \~\@nameuse{MT@\MT@feat @inh@name}'}%
1544 </debug>
1545 \MT@let@cn\@tempc{MT@\MT@feat @inh\@csname MT@\MT@feat @inh@name\endcsname}%

```

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

```

1546 \ifx\@tempc\@empty \else
1547 <debug> \MT@dinfo@n1{1}{parsing inheritance list ...}%
1548 \MT@let@cn\MT@inh@name{MT@\MT@feat @inh@name}%
1549 \edef\MT@curr@list@name{inheritance list\MessageBreak~\MT@inh@name}'%

```

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

```

1550 \begingroup
1551 \def\MT@cat@inh{%
1552 \MT@set@inputenc
1553 \expandafter\MT@inh@do\@tempc,\relax,%
1554 \global\MT@let@nc{MT@\MT@feat @inh\@csname MT@\MT@feat @inh@name\endcsname}\@empty

```

```

1555     \endgroup
1556     \fi
1557   }{%
1558     \MT@let@nc{MT@MT@feat @inh@name}\@undefined
1559   }%
1560 }

```

### 13.4.8 Translating Characters into Slots

Get the slot number of the character in the current encoding.

`\MT@get@slot` There are lots of possibilities how a character may be specified in the configuration files, which makes translating them into slot numbers quite expensive. Also, we want to have this as robust as possible, so that the user does not have to solve a sphinx's riddle if anything goes wrong.

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

```

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

```

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

```

1565   \MT@toks=\expandafter{\@tempa}%

```

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

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

```

1566     \expandafter\MT@is@letter\@tempa\relax\relax
1567     \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.

```

1568     \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  $\text{\LaTeX}$ '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 `\'i` or `\U\CYRI`, hence, `\string` wouldn't be safe enough.

```

1569     \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
1570     \MT@is@symbol

```

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

```

1571     {\expandafter\MT@is@composite\@tempa\relax\relax}%
1572     \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.

```

1573     \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1574     \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
1575     \fi
1576     \fi

```

```

1577 \let\MT@char\MT@char@
1578 \ifnum\MT@char < \z@
1579 \MT@warn@unknown
1580 \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.

```

1581 \ifMT@noreset \else
1582 \MT@warn@reset
1583 \let\MT@char\m@ne
1584 \fi
1585 \fi
1586 \escapechar\m@ne
1587 }

```

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

```

\MT@testrest 1588 \newif\ifMT@noreset
1589 \def\MT@testrest#1#2{\MT@ifstreq{#1}{#2}\relax\MT@noresetfalse}

```

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

```

1590 \def\MT@is@letter#1#2\relax{%
1591 \ifcat a\noexpand#1\relax
1592 \edef\MT@char@{\number`#1}%
1593 \ifx\#2\%
1594 <debug>\MT@dinfo@n1{3}{> `the\MT@toks' is a letter (\MT@char@)}%
1595 \else
1596 \MT@noresetfalse
1597 \fi
1598 \else
1599 \ifcat !\noexpand#1\relax
1600 \edef\MT@char@{\number`#1}%
1601 <debug>\MT@dinfo@n1{3}{> `the\MT@toks' is a character (\MT@char@)}%
1602 \ifx\#2\%
1603 \ifnum\MT@char@>127 \MT@warn@ascii \fi
1604 \else
1605 \MT@noresetfalse
1606 \expandafter\MT@is@number#1#2\relax\relax
1607 \fi
1608 \fi
1609 \fi
1610 }

```

`\MT@is@number` Numbers may be specified as a three-digit decimal number (029), as a hexadecimal number (prefixed with “: ”1D) or as an octal number (prefixed with ‘: ’35). They must consist of at least three characters (including the prefix), that is, “F is not permitted.

```

1611 \def\MT@is@number#1#2#3\relax{%
1612 \ifx\relax#3\relax \else
1613 \ifx\relax#2\relax \else
1614 \MT@noresettrue
1615 \if#1"\relax
1616 \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
1617 <debug>\MT@dinfo@n1{3}{> ... a hexadecimal number: \MT@char@}%
1618 \else
1619 \if#1'\relax
1620 \def\MT@char@{\number#1#2#3}%
1621 <debug>\MT@dinfo@n1{3}{> ... an octal number: \MT@char@}%
1622 \else
1623 \MT@ifint{#1#2#3}{%
1624 \def\MT@char@{\number#1#2#3}%

```

```

1625 <debug>\MT@info{n1{3}> ... a decimal number: \MT@char}%
1626         }\MT@noestfalse
1627         \fi
1628         \fi
1629         \ifnum\MT@char > \@cclv
1630             \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
1631             \let\MT@char@m@ne
1632         \fi
1633     \fi
1634 \fi
1635 }

```

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

Previous solution, slightly more robust (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@inpec\the\toks@\relax\relax
      \edef\@tempa{\the\toks@}%
      \edef\x{\MT@toks={\the\MT@toks\space(= \the\toks@)}}\x
    \fi
  \fi
}
\def\MT@active@inpec#1#2#3\relax{%
  \ifx#1\IeC
    \def\IeC##1{\toks@={##1}}%
    \the\toks@
    \expandafter\MT@active@inpec\the\toks@\relax\relax
  \fi
  \ifx#1\@tabacckludge
    \def\@tabacckludge##1##2{%
      \toks@=\expandafter{\csname\string##1\endcsname#3}}%
    \the\toks@
  \fi
  \ifx#1\@inpec@undefined@
    \def\@inpec@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.

```
1636 \def\MT@is@active#1#2\@nil{%
1637   \ifnum\catcode`#1 = \active
1638     \begingroup
1639     \set@display@protect
1640     \let\IeC\@firstofone
1641     \let\@inpenc@undefined@\MT@undefined@char
```

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

```
1642   \def\UTFviii@defined##1{\ifx ##1\relax
1643     \MT@undefined@char{utf8}\else\expandafter##1\fi}%
1644   \edef\x{%
1645     \def\noexpand\@tempa{\@tempa}%
```

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

```
1646     \MT@toks={\the\MT@toks\space(= \@tempa)}%
1647   }%
1648   \expandafter\endgroup\x
1649 \fi
1650 }
```

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

```
1651 \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.

```
1652 \def\MT@is@symbol{%
1653   \edef\MT@char{\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
1654   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1655     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
1656   \ifnum\MT@char@ < \z@
```

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

```
1657   \expandafter\MT@is@letter\MT@char\relax\relax
1658 \fi
1659 }
```

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

```
\MT@charstring 1660 \begingroup
1661   \catcode`\/=0
1662   /MT@map@tlist@n{/C/H/A/R}/@makeother
1663   /lowercase{%
1664     /def/x{%
1665       /def/MT@charstring{\CHAR"%
1666       /def/MT@is@char##1\CHAR"##2##3##4/relax{%
1667         /ifx/relax##1/relax
1668         /if##3\relax
1669         /edef/MT@char@{/number"##2}%
1670         /MT@testrest/MT@charstring{##3##4}%
1671       /else
1672         /edef/MT@char@{/number"##2##3}%
1673         /MT@testrest/MT@charstring{##4}%
1674       /fi
1675   <debug> /MT@dinfo@n1{3}{> `~/the/MT@toks' is a \char (/MT@char@)}%
1676   /fi
```

```

1677     }%
1678   }%
1679 }
1680 /expandafter/endgroup/x

```

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

```

1681 \def\MT@is@composite#1#2\relax{%
1682   \ifx\#2\else

```

Again, we construct a control sequence, this time of the form: `\langle encoding \rangle \langle accent \rangle - \langle character \rangle`, 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 `\stringify`ing it. Thus, we will die gracefully even on wrong Unicode input without `utf8`.

```

1683   \edef\MT@char{\expandafter
1684     \csname\expandafter
1685       \string\csname\MT@encoding\endcsname
1686       \MT@detokenize@n{#1}-%
1687       \MT@detokenize@n{#2}%
1688     \endcsname}%
1689   \expandafter\MT@is@letter\MT@char\relax\relax
1690 \fi
1691 }

```

(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 1692 \def\MT@set@listname{%
1693   \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list\MessageBreak
1694     \@nameuse{MT@MT@feat @c@name}'}%
1695 }

```

`\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.

```

1696 \def\MT@warn@ascii{%
1697   \MT@warning@n1{Character `the\MT@toks' (= \MT@char@)
1698     is outside of ASCII range.\MessageBreak
1699   You must load the `inputenc' package before using\MessageBreak
1700   8-bit characters in \MT@curr@list@name}%

```



```

1701 }
\MT@warn@number@too@large Number too large.
1702 \def\MT@warn@number@too@large#1{%
1703   \MT@warning@n1{%
1704     Number #1 in encoding `\'MT@encoding' too large!\MessageBreak
1705     Ignoring it in \MT@curr@list@name}%
1706 }

\MT@warn@rest Not all of the string has been parsed.
1707 \def\MT@warn@rest{%
1708   \MT@warning@n1{%
1709     Unknown slot number of character\MessageBreak`\'the\MT@toks'\MessageBreak
1710     in font encoding `\'MT@encoding'\MT@warn@maybe@inputenc
1711     Make sure it's a single character\MessageBreak
1712     (or a number) in \MT@curr@list@name}%
1713 }

\MT@warn@unknown No idea what went wrong.
1714 \def\MT@warn@unknown{%
1715   \MT@warning@n1{%
1716     Unknown slot number of character\MessageBreak`\'the\MT@toks'\MessageBreak
1717     in font encoding `\'MT@encoding'\MT@warn@maybe@inputenc
1718     in \MT@curr@list@name}%
1719 }

\MT@warn@maybe@inputenc In case an input encoding had been requested.
1720 \def\MT@warn@maybe@inputenc{%
1721   \MT@ifdefined@n@T
1722   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
1723   { (input encoding `\'%
1724     \@nameuse
1725     {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')%
1726     \MessageBreak}%
1727 }

```

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

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

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

For my reference:

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

- `\get@external@font` is called by `\extract@font`, by itself, and by the substitution macros

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

```
1728 \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.

```
1729 \let\MT@font\@empty
```

```
1730 \MT@addto@setup{%
```

```
1731   \g@addto@macro\do@subst@correction{%
```

```
1732     \xdef\MT@font{\csname \curr@fontshape/\f@size\endcsname}%
```

```
1733   }
```

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

```
1734 \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
```

```
1735 \ifx\pickup@font\MT@orig@pickupfont \else
```

```
1736   \MT@warning@nl{%
```

```
1737     Command \string\pickup@font\space is not defined as expected.\MessageBreak
```

```
1738     Double-check whether micro-typography is indeed.\MessageBreak
```

```
1739     applied to the document.\MessageBreak (Hint: Turn on `verbose' mode)%
```

```
1740   }
```

```
1741 \fi
```

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

```
1742 \g@addto@macro\pickup@font{\begingroup}
```

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

```
1743 \MT@with@package{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
```

```
1744 \g@addto@macro\pickup@font{%
```

```
1745   \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.

```
1746   \ifx\MT@font\@empty
```

```
1747     \let\MT@font\font@name
```

```
1748   \else
```

```

1749     \ifx\MT@font\font@name \else
1750     \MT@register@subst@font
1751     \fi
1752     \fi

```

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

```
1753     \MT@check@font
```

If not, set it up and register it.

```

1754     \ifMT@inlist@ \else
1755     \MT@setupfont
1756     \MT@register@font
1757     \fi
1758     \global\let\MT@font\@empty
1759     \endgroup
1760     }

```

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

```
1761     \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, L<sup>A</sup>T<sub>E</sub>X 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.

```

1762     \let\MT@orig@add@accent\add@accent
1763     \def\add@accent#1#2{%
1764     \let\pickup@font\MT@orig@pickupfont
1765     \MT@orig@add@accent{#1}{#2}%
1766     \let\pickup@font\MT@pickupfont
1767     }
1768 }

```

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.

```
1769 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

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

```
1770 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
```

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

```
1771 \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.

```
1772 \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.

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

```

1773 \def\MT@check@font@cx{%
1774   \@tempswattrue
1775   \MT@map@clist@cx\MT@active@features{%
1776     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
1777     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1778     \ifMT@inlist@
1779       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
1780     \else
1781       \@tempswafalse
1782     \fi
1783   }%
1784   \if@tempswa \MT@inlist@true \else \MT@inlist@false \fi
1785 }

```

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

```

1786 \def\MT@register@subst@font@cx{%
1787   \MT@map@clist@cx\MT@active@features{%
1788     \expandafter\MT@xadd
1789     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1790     {\font@name,}%
1791   }%
1792 }

```

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

```

1793 \def\MT@register@font@cx{%
1794   \MT@map@clist@cx\MT@active@features{%
1795     \expandafter\ifx\csname MT@\@nameuse{MT@abbr@##1}\endcsname\relax\else
1796     \expandafter\MT@xadd
1797     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1798     {\MT@font,}%
1799     \def\@tempa{##1}%
1800     \expandafter\MT@map@tlist@cx
1801     \csname MT@##1@doc@contexts\endcsname
1802     \MT@rem@from@lists
1803   \fi
1804 }%
1805 }

```

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

```

1806 \def\MT@rem@from@lists#1{%
1807   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
1808     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist
1809     \expandafter\MT@font\csname MT@\@tempa @#1font@list\endcsname
1810   }%
1811 }

```

`\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.

```

1812 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
1813 \MT@addto@setup{%
1814   \def\microtypecontext#1{%
1815     \MT@setup@contexts
1816     \setkeys{MTC}{#1}%
1817     \selectfont
1818     \aftergroup\MT@reset@context
1819   }%
1820 }

```

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

```

1821 \def\MT@reset@context{%
1822   \MT@vinfo{Resetting contexts on line \the\inputlineno
1823   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
1824   <debug&beta>           /\MT@kn@context/\MT@sp@context
1825   }%
1826   \selectfont
1827 }

```

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

```

1828 \def\MT@setup@contexts{%
1829   \MT@map@clist@c\MT@active@features
1830   {\global\let\MT@let@nc\MT@#1@font@list}\MT@font@list}%
1831   \global\let\MT@check@font\MT@check@font@cx
1832   \global\let\MT@register@font\MT@register@font@cx
1833   \global\let\MT@register@subst@font\MT@register@subst@font@cx
1834   \global\let\MT@setup@contexts\relax
1835 }

```

`\MT@define@context`

```

1836 \def\MT@define@context#1{%
1837   \define@key{MTC}{#1}[]{}%
1838   \KV@sp@def\@tempb{#1}%
1839   \edef\@tempb{\@nameuse{MT@rbba@\@tempb}}%
1840   \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
1841   \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).

```

1842   \MT@ifempty{##1}{\def\MT@val{@}}{\KV@sp@def\MT@val{##1}}%
1843   \expandafter\ifx\csname MT@\@tempb @context\endcsname\MT@val \else
1844   \MT@vinfo{--- Changing #1 context to `\'MT@val'}%

```

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

```

1845   \global\MT@let@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes}%

```

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

```

1846   \expandafter\MT@exp@one@n\expandafter\MT@in@tlist
1847   \expandafter\MT@val\csname MT@\@tempb @doc@contexts\endcsname
1848   \ifMT@inlist@ \else
1849   \expandafter\MT@xadd\csname MT@\@tempb @doc@contexts\endcsname{{\MT@val}}%
1850   <debug> \MT@dinfo{2}{>>> added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
1851   \fi
1852   \MT@edef{n{MT@\@tempb @context}}{\MT@val}%
1853   \fi
1854   \fi
1855   }%
1856 }

1857 \MT@map@clist@c\MT@features@long{\MT@define@context{#1}}

```

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

```

\MT@ex@context 1858 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%

```

```

\MT@sp@context 1859 \MT@def@n{MT@#1@context}{@}%

```

```

\MT@kn@context 1860 \MT@def@n{MT@#1@doc@contexts}{@}%

```

```

1861 }

```

```

\MT@pr@doc@contexts 1862 \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.

```
1863 \def\DeclareMicrotypeSet{%
1864   \ifstar
1865     {\ifnextchar[\MT@DeclareSetAndUseIt
1866       {\MT@DeclareSetAndUseIt[]}}%
1867     {\ifnextchar[\MT@DeclareSet
1868       {\MT@DeclareSet[]}}%
1869 }
```

```
\MT@DeclareSet
\MT@DeclareSetAndUseIt 1870 \def\MT@DeclareSet[#1]{%
1871   \MT@DeclareSet@{#1}%
1872 }
1873 \def\MT@DeclareSetAndUseIt[#1]#2#3{%
1874   \MT@DeclareSet@{#1}{#2}{#3}%
1875   \UseMicrotypeSet[#1]{#2}%
1876 }
```

```
\MT@DeclareSet@
1877 \def\MT@DeclareSet@#1#2#3{%
1878   \KV@sp@def\@tempa{#1}%
1879   \MT@ifempty\@tempa{%
1880     \MT@map@clistc\MT@features{\MT@declare@sets{##1}{#2}{#3}}}%
1881   }{%
1882     \MT@map@clistc\@tempa{%
1883       \KV@sp@def\@tempa{##1}%
1884       \MT@ifempty\@tempa\relax{%
1885         \MT@is@feature{set declaration `#2'}{%
1886           \MT@exp@one@n\MT@declare@sets
1887             {\csname MT@rbb@{\@tempa\endcsname}{#2}{#3}%
1888           }%
1889         }%
1890       }%
1891     }%
1892 }
```

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

```
1893 \let\MT@curr@set@name\empty
```

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

```
1894 \def\MT@declare@sets#1#2#3{%
1895   \KV@sp@def\MT@curr@set@name{#2}%
1896   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
1897     \MT@warning{Redefining set `~\MT@curr@set@name'}%
1898   }%
1899   \global\MT@let@nc{MT@#1@set@@\MT@curr@set@name}\empty
1900   <debug>\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
1901   \setkeys{MT@#1@set}{#3}%
1902 }
```

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

```

1903 \def\MT@define@set@keys#1{%
1904 \MT@define@set@key@{encoding}{#1}%
1905 \MT@define@set@key@{family}{#1}%
1906 \MT@define@set@key@{series}{#1}%
1907 \MT@define@set@key@{shape}{#1}%
1908 \MT@define@set@key@size{#1}%
1909 \MT@define@set@key@font{#1}%
1910 }

```

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

```

1911 \def\MT@define@set@key@#1#2{%
1912 \define@key{MT@#2@set}{#1}[]{%
1913 \global\MT@let@enc{MT@#2list@#1@MT@curr@set@name}\@empty
1914 \MT@map@clist@n{##1}{%
1915 \KV@sp@def\MT@val{###1}%
1916 \MT@get@highlevel{#1}%

```

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

```

1917 \MT@exp@two@n@g@addto@macro
1918 {\csname MT@#2list@#1@MT@curr@set@name\expandafter\endcsname}%
1919 {\MT@val,}%
1920 }%

```

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

```

1921 \edef\x{%
1922 \expandafter\noexpand\expandafter\MT@fix@font@spec
1923 \expandafter\noexpand\csname MT@#2list@#1@MT@curr@set@name\endcsname
1924 }%
1925 \MT@exp@one@n\MT@addto@setup\x
1926 <debug>\MT@info@n1{1}{-- #1: \nameuse{MT@#2list@#1@MT@curr@set@name}}%
1927 }%
1928 }

```

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

```

1929 \def\MT@get@highlevel#1{%
1930 \expandafter\MT@test@ast\MT@val*\@nil{%

```

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

```

1931 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
1932 \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.

```

1933 }%
1934 }

```

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

```

1935 \def\MT@test@ast#1*#2\@nil{%
1936 \def\@tempa{#1}%
1937 \MT@ifempty{#2}\@gobble\@firstofone
1938 }

```

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

```

1939 \def\MT@fix@font@spec#1{%
1940 \xdef#1{#1}%
1941 \global\MT@make@string#1%
1942 }

```

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

```

1943 \def\MT@define@set@key@size#1{%
1944 \define@key{MT@#1@set}{size}[]{%

```

```

1945 \MT@map@clist@n{##1}{%
1946 \KV@sp@def\MT@val{###1}%
1947 \expandafter\MT@get@range\MT@val--\@nil
1948 \ifx\MT@val\relax \else
1949 \expandafter\MT@xadd
1950 \csname MT@#1list@size@\MT@curr@set@name\endcsname
1951 {{{\MT@lower}{\MT@upper}\relax}}%
1952 \fi
1953 }%
1954 <debug>\MT@dinfo@n1{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
1955 }%
1956 }

```

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 1957 \def\MT@get@range#1-#2-#3@nil{%
1958 \MT@ifempty{#1}{%
1959 \MT@ifempty{#2}{%
1960 \let\MT@val\relax
1961 }{%
1962 \def\MT@lower{0}%
1963 \def\MT@val{#2}%
1964 \MT@get@size
1965 \edef\MT@upper{\MT@val}%
1966 }%
1967 }{%
1968 \def\MT@val{#1}%
1969 \MT@get@size
1970 \ifx\MT@val\relax \else
1971 \edef\MT@lower{\MT@val}%
1972 \MT@ifempty{#2}{%
1973 \MT@ifempty{#3}{%
1974 {\def\MT@upper{-1}}%

```

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

```

1975 {\def\MT@upper{2048}}%
1976 }%
1977 \def\MT@val{#2}%
1978 \MT@get@size
1979 \ifx\MT@val\relax \else
1980 \MT@ifdim\MT@lower>\MT@val{%
1981 \MT@warning{%
1982 Invalid size range (\MT@lower\space > \MT@val) in font set
1983 ~\MT@curr@set@name'.\MessageBreak Swapping sizes%
1984 \edef\MT@upper{\MT@lower}%
1985 \edef\MT@lower{\MT@val}%
1986 }{%
1987 \edef\MT@upper{\MT@val}%
1988 }%
1989 \MT@ifdim\MT@lower=\MT@upper
1990 {\def\MT@upper{-1}}%
1991 \relax
1992 \fi
1993 }%
1994 \fi

```



```
1995 }%
1996 }
```

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

```
1997 \def\MT@get@size{%
```

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

```
1998 \if*\MT@val\relax
1999 \def\@tempa{\normalsize}%
2000 \else
2001 \MT@let@cn\@tempa{\MT@val}%
2002 \fi
2003 \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).

```
2004 \begingroup
2005 \def\set@fontsize##1##2##3##4\@nil{\gdef\MT@val{##2}}%
2006 \@tempa\@nil
2007 \endgroup
2008 \fi
```

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

```
2009 \MT@ifdimen\MT@val{%
2010 \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2011 \edef\MT@val{\strip@pt\@tempdima}%
2012 }{%
2013 \MT@warning{Could not parse font size `~\MT@val'~\MessageBreak
2014 in font set `~\MT@curr@set@name'~}%
2015 \let\MT@val\relax
2016 }%
2017 }
```

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

```
2018 \def\MT@define@set@key@font#1{%
2019 \define@key{MT@#1@set}{font}[]{}%
2020 \global\MT@let@nc{MT@#1list@font}\MT@curr@set@name}\@empty
2021 \MT@map@clist@n{##1}{%
2022 \KV@sp@def\MT@val{####1}%
2023 \expandafter\MT@get@font\MT@val///// \@nil
2024 \MT@exp@two@n@g@addto@macro
2025 {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
2026 {\MT@val,}%
2027 }%
2028 \edef\x{%
2029 \expandafter\noexpand\expandafter\MT@fix@font@spec
2030 \expandafter\noexpand\csname MT@#1list@font@\MT@curr@set@name\endcsname
2031 }%
2032 \MT@exp@one@n\MT@addto@setup\x
2033 (debug) \MT@dinfo@n1{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
2034 }%
2035 }
```

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

```
2036 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2037 \MT@ifempty{#1#2#3#4#5}\relax{%
```

```

2038 \let\@tempb\@empty
2039 \def\MT@temp{#1/#2/#3/#4/#5}%
2040 \MT@get@axis{encoding}{#1}%
2041 \MT@get@axis{family}{#2}%
2042 \MT@get@axis{series}{#3}%
2043 \MT@get@axis{shape}{#4}%
2044 \MT@ifempty{#5}{%
2045   \MT@warn@axis@empty{size}{\string\normalsize}%
2046   \def\MT@val{*}%
2047 }{%
2048   \def\MT@val{#5}%
2049 }%
2050 \MT@get@size
2051 \ifx\MT@val\relax\def\MT@val{0}\fi
2052 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2053 \let\MT@val\@tempb
2054 }%
2055 }

```

\MT@get@axis

```

2056 \def\MT@get@axis#1#2{%
2057   \def\MT@val{#2}%
2058   \MT@get@highlevel{#1}%
2059   \MT@ifempty\MT@val{%
2060     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2061     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2062   }\relax
2063   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2064 }

```

\MT@warn@axis@empty

```

2065 \def\MT@warn@axis@empty#1#2{%
2066   \MT@warning{#1 axis is empty in font specification\MessageBreak
2067     ~\MT@temp'. Using ~#2' instead}%
2068 }

```

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

```
2069 \MT@map@clist@c\MT@features{\MT@define@set@keys{#1}}
```

It is also used for \DisableLigatures.

```
2070 \MT@define@set@keys{nl}
```

\UseMicrotypeSet To use a particular set we simply redefine MT@(*feature*)@setname. If the optional argument is empty, set names for all features will be redefined.

```

2071 \renewcommand*\UseMicrotypeSet[2][]{%
2072   \KV@@sp@def\@tempa{#1}%
2073   \MT@ifempty\@tempa{%
2074     \MT@map@clist@c\MT@features{\MT@use@set{##1}{#2}}%
2075   }{%
2076     \MT@map@clist@c\@tempa{%
2077       \KV@@sp@def\@tempa{#1}%
2078       \MT@ifempty\@tempa\relax{%
2079         \MT@is@feature{activation of set ~#2'}{%
2080           \MT@exp@one@n\MT@use@set
2081             {\csname MT@rbb@\@tempa\endcsname}{#2}%
2082         }%
2083       }%
2084     }%
2085   }%
2086 }

```

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

\MT@ex@setname

\MT@use@set

```

2087 \def\MT@use@set#1#2{%
2088 \KV@sp@def\@tempa{#2}%
2089 \MT@ifdefined@n@TF{MT@#1@set@\@tempa}{%
2090 \global\MT@edef@n{MT@#1@setname}{\@tempa}%
2091 \MT@info{Using \@nameuse{MT@abbr@#1} set '\@tempa'}%
2092 }{%
2093 \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2094 \global\MT@edef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2095 }%
2096 \MT@warning{%
2097 The \@nameuse{MT@abbr@#1} set '\@tempa' is undeclared.\MessageBreak
2098 Using set '\@nameuse{MT@#1@setname}' instead}%
2099 }%
2100 }

```

`\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'.

```

2101 \MT@requires@pdftex5{
2102 \renewcommand*\DisableLigatures[1]{%
2103 \edef\MT@active@features{\MT@active@features,nl}%
2104 \MT@noligaturestrue
2105 \MT@declare@sets{nl}{no ligatures}{#1}%
2106 \gdef\MT@n@setname{no ligatures}%
2107 }
2108 }{

```

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

```

2109 \renewcommand*\DisableLigatures[1]{%
2110 \MT@warning{Disabling ligatures of a font is only possible.\MessageBreak
2111 with pdftex version 1.30 or newer.\MessageBreak
2112 Ignoring \string\DisableLigatures}%
2113 \global\let\DisableLigatures\@gobble
2114 }
2115 }

```

`\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.

```

2116 \renewcommand*\DeclareMicrotypeSetDefault[2][ ]{%
2117 \KV@sp@def\@tempa{#1}%
2118 \MT@ifempty\@tempa{%
2119 \MT@map@clist@c\MT@features{\MT@set@default@set{##1}{#2}}%
2120 }{%
2121 \MT@map@clist@c\@tempa{%
2122 \KV@sp@def\@tempa{#1}%
2123 \MT@ifempty\@tempa\relax{%
2124 \MT@is@feature{declaration of default set `#2'}{%
2125 \MT@exp@one@n\MT@set@default@set
2126 {\csname MT@rbb@@\@tempa\endcsname}{#2}%
2127 }%
2128 }%
2129 }}%
2130 }%
2131 }

```

`\MT@default@pr@set`

`\MT@default@ex@set` 2132 \def\MT@set@default@set#1#2{%

`\MT@default@kn@set` 2133 \KV@sp@def\@tempa{#2}%

`\MT@default@sp@set` 2134 \MT@ifdefined@n@TF{MT@#1@set@\@tempa}{%

`\MT@set@default@set` 2135 *<debug>* \MT@info{1}{declaring default \@nameuse{MT@abbr@#1} set '\@tempa'}%

2136 \global\MT@edef@n{MT@default@#1@set}{\@tempa}%

2137 }%

2138 \MT@warning{%

```

2139     The \@nameuse{MT@abbr@#1} set '\@tempa' is not declared.\MessageBreak
2140     Cannot make it the default set. Using set\MessageBreak `all' instead}%
2141     \global\MT@edef@n{MT@default@#1@set}{all}%
2142 }%
2143 }

```

`\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.

```

2144 \renewcommand*\DeclareMicrotypeAlias[2]{%
2145   \KV@sp@def\@tempa{#1}%
2146   \KV@sp@def\@tempb{#2}%
2147   \MT@make@string\@tempb
2148   \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2149     \MT@warning{Alias font family '\@tempb' will override
2150     alias '\@nameuse{MT@\@tempa @alias}'\MessageBreak
2151     for font family '\@tempa'}}%
2152   \global\MT@edef@n{MT@\@tempa @alias}{\@tempb}%

```

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

```

2153   \MT@ifdefined@c@T{MT@family}{%
2154   <debug>\MT@dinfo{1}{Activating alias font '\@tempb' for '\MT@family'}%
2155     \global\let\MT@familyalias\@tempb
2156 }%
2157 }

```

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

```

2158 \def\LoadMicrotypeFile#1{%
2159   \KV@sp@def\@tempa{#1}%
2160   \MT@make@string\@tempa
2161   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2162   \ifMT@inlist@
2163     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2164   \else
2165     \MT@xadd\MT@file@list{\@tempa,}%
2166     \MT@begin@catcodes
2167     \InputIfFileExists{mt-\@tempa.cfg}{%
2168       \edef\MT@curr@file{mt-\@tempa.cfg}%
2169       \MT@vinfo{... Loading configuration file \MT@curr@file}%
2170     }{%
2171       \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2172       does not exist}%
2173     }%
2174   \MT@end@catcodes
2175   \fi
2176 }

```

### 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.

```

2177 <beta>
2178 \def\DeclareMicrotypeBabelHook#1#2{%
2179   \MT@map@clist@n{#1}{%
2180     \KV@sp@def\@tempa{##1}%
2181     \global\MT@def@n{MT@babel@\@tempa}{#2}%
2182   }%
2183 }
2184 </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
\MT@pr@c@name character protrusion factors.
\MT@extra@load A new macro called \MT@pr@c@<name> will be defined to be <#3> (i. e., the list
\MT@extra@factor of characters, not expanded).
\MT@extra@preset 2185 \renewcommand*\SetProtrusion[2] [] {%
\MT@extra@unit 2186 \let\MT@pr@c@name\undefined
\MT@extra@context 2187 \let\MT@extra@context\empty
\MT@extra@context 2188 \MT@map@clist@n{load,factor,unit,preset,inputenc}{\MT@let@nc{\MT@extra@##1}\undefined}%
\MT@permutelist Parse the optional first argument:
2189 \setkeys{MT@pr@c}{#1}%

If the user hasn't specified a name, we will create one.
2190 \MT@get@codes@name{pr}%

Extra options.
2191 \MT@map@clist@n{factor,unit,preset,inputenc}{\MT@set@opt{pr}{##1}}%
2192 (debug) \MT@info{1}{creating protrusion list ~\MT@pr@c@name'}%
2193 \def\MT@permutelist{prc}%
2194 \setkeys{MT@cfg}{#2}%

We have parsed the second argument, and can now define macros for all permuta-
tions of the font attributes to point to \MT@pr@c@<name>, ...
2195 \MT@permute

... which we can now define to be <#3>. We want the catcodes to be correct even
if this is called in the preamble.
2196 \MT@begin@catcodes
2197 \MT@set@pr@list
2198 }

\MT@set@pr@list Here, as elsewhere, we have to make the definitions global, since they will occur
inside a group.
2199 \def\MT@set@pr@list#1{%
2200 \global\MT@def@n{\MT@pr@c@\MT@pr@c@name}{#1}%
2201 \MT@end@catcodes
2202 }

\SetExpansion \SetExpansion only differs in that it allows some extra options (stretch, shrink,
\MT@ex@c@name step, auto).
\MT@extra@load 2203 \renewcommand*\SetExpansion[2] [] {%
\MT@extra@factor 2204 \let\MT@ex@c@name\undefined
\MT@extra@factor 2205 \let\MT@extra@context\empty
\MT@extra@stretch 2206 \MT@map@clist@n{load,factor,preset,inputenc,stretch,shrink,step,auto}%
\MT@extra@shrink 2207 {\MT@let@nc{\MT@extra@##1}\undefined}%
\MT@extra@step 2208 \setkeys{MT@ex@c}{#1}%
\MT@extra@auto 2209 \MT@get@codes@name{ex}%

\MT@extra@preset The extra options to \SetExpansion also have to be dealt with only after we know
\MT@extra@context the name.
\MT@extra@inputenc 2210 \MT@map@clist@n{preset,inputenc,stretch,shrink,step,auto}{\MT@set@opt{ex}{##1}}%
2211 \MT@ifdefined@cT\MT@extra@factor{%
2212 \ifnum\MT@extra@factor>\@m
2213 \MT@warning@n1{Expansion factor \number\MT@extra@factor\space too
2214 large in list\MessageBreak ~\MT@ex@c@name'. Setting it to the

```

```

2215         maximum of 1000}%
2216     \let\MT@extra@factor\@m
2217     \fi
2218     \global\MT@let@nc{MT@ex@c@\MT@ex@c@name @factor}\MT@extra@factor
2219 }%
2220 (debug)\MT@dinfo{1}{creating expansion list ~\MT@ex@c@name'}%
2221 \def\MT@permutelist{ex@c}%
2222 \setkeys{MT@cfg}{#2}%
2223 \MT@permute
2224 \MT@begin@catcodes
2225 \MT@set@ex@list
2226 }

\MT@set@ex@list Same story.
2227 \def\MT@set@ex@list#1{%
2228   \global\MT@def@n{MT@ex@c@\MT@ex@c@name}{#1}%
2229   \MT@end@catcodes
2230 }
2231 (*beta)

\SetExtraSpacing
  \MT@sp@c@name 2232 \renewcommand*\SetExtraSpacing[2] [] {%
  \MT@extra@load 2233   \let\MT@sp@c@name\undefined
  \MT@extra@context 2234   \let\MT@extra@context\@empty
  \MT@extra@factor 2235   \MT@map@clist@n{load,factor,unit,preset,inputenc}{\MT@let@nc{MT@extra@##1}\undefined}%
  \MT@extra@unit 2236   \setkeys{MT@sp@c}{#1}%
  \MT@extra@preset 2237   \MT@get@codes@name{sp}%
  \MT@extra@context 2238   \MT@map@clist@n{factor,unit,preset,inputenc}{\MT@set@opt{sp}{##1}}%
  \MT@extra@inputenc 2239 (debug)\MT@dinfo{1}{creating spacing list ~\MT@sp@c@name'}%
  \MT@extra@inputenc 2240   \def\MT@permutelist{sp@c}%
  2241   \setkeys{MT@cfg}{#2}%
  2242   \MT@permute
  2243   \MT@begin@catcodes
  2244   \MT@set@sp@list
  2245 }

\MT@set@sp@list
2246 \def\MT@set@sp@list#1{%
2247   \global\MT@def@n{MT@sp@c@\MT@sp@c@name}{#1}%
2248   \MT@end@catcodes
2249 }

\SetExtraKerning
  \MT@kn@c@name 2250 \renewcommand*\SetExtraKerning[2] [] {%
  \MT@extra@load 2251   \let\MT@kn@c@name\undefined
  \MT@extra@context 2252   \let\MT@extra@context\@empty
  \MT@extra@factor 2253   \MT@map@clist@n{load,factor,unit,preset,inputenc}{\MT@let@nc{MT@extra@##1}\undefined}%
  \MT@extra@unit 2254   \setkeys{MT@kn@c}{#1}%
  \MT@extra@preset 2255   \MT@get@codes@name{kn}%
  \MT@extra@context 2256   \MT@map@clist@n{factor,unit,preset,inputenc}{\MT@set@opt{kn}{##1}}%
  \MT@extra@inputenc 2257 (debug)\MT@dinfo{1}{creating kerning list ~\MT@kn@c@name'}%
  \MT@extra@inputenc 2258   \def\MT@permutelist{kn@c}%
  2259   \setkeys{MT@cfg}{#2}%
  2260   \MT@permute
  2261   \MT@begin@catcodes
  2262   \MT@set@kn@list
  2263 }

\MT@set@kn@list
2264 \def\MT@set@kn@list#1{%
2265   \global\MT@def@n{MT@kn@c@\MT@kn@c@name}{#1}%

```

```
2266 \MT@end@catcodes
2267 }
```

```
2268 </beta>
```

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

```
2269 \def\MT@get@codes@name#1{%
2270 \MT@ifdefined@n@TF{MT@#1@c@name}{%
2271 \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
2272 \MT@warning{Redefining list `\<@nameuse{MT@#1@c@name}'}}%
2273 }%
2274 }{%
2275 \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2276 }%
2277 \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
```

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

```
2278 \MT@ifdefined@c@T\MT@extra@load{%
2279 \global\MT@let@nc{MT@#1@c@\MT@curr@set@name load}\MT@extra@load
2280 }%
2281 }
```

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

```
2282 \def\MT@set@opt#1#2{%
2283 \MT@ifdefined@n@T{MT@extra@#2}{%
2284 \global\MT@let@nn{MT@#1@c@\csname MT@#1@c@name\endcsname @#2}{MT@extra@#2}%
2285 }%
2286 }
```

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

```
2287 \def\MT@define@code@key#1#2{%
2288 \define@key{MT@#2}{#1}[]{%
2289 \@tempcnta=\@ne
2290 \MT@map@clist@n{##1}{%
2291 \KV@sp@def\MT@val{###1}%
```

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

```
2292 \MT@get@highlevel{#1}%
2293 \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
2294 \advance\@tempcnta \@ne
2295 }%
2296 }%
2297 }
```

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

```
2298 \def\MT@define@code@key@size#1{%
2299 \define@key{MT@#1}{size}[]{%
2300 \MT@map@clist@n{##1}{%
2301 \KV@sp@def\MT@val{###1}%
2302 \expandafter\MT@get@range\MT@val--\@nil
2303 \ifx\MT@val\relax \else
2304 \expandafter\MT@add\csname MT@temp@size\endcsname
2305 {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2306 \fi
2307 }%
2308 }%
2309 }
```

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

```
2310 \def\MT@define@code@key@font#1{%
```

```

2311 \define@key{MT@#1}{font}[]{%
2312 \MT@map@clist@n{##1}%
2313 \KV@sp@def\MT@val{###1}%
2314 \expandafter\MT@get@font@and@size\MT@val/////\/\nil
2315 \global\MT@edef@n{MT@MT@permutelist @\@tempb\MT@extra@context}%
2316 {\csname MT@MT@permutelist @name\endcsname}%
2317 *debug
2318 \MT@dinfo@n1{1}{initializing: use list for font \@tempb=\MT@val
2319 \ifx\MT@extra@context\@empty\else\MessageBreak
2320 (context: \MT@extra@context)\fi}%
2321 /debug
2322 \expandafter\MT@xaddb
2323 \csname MT@MT@permutelist @\@tempb\MT@extra@context @sizes\endcsname
2324 {{{\MT@val}\m@ne}\MT@curr@set@name}}}%
2325 }%
2326 }%
2327 }

```

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

```

2328 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2329 \MT@ifempty{#1#2#3#4#5}\relax{%
2330 \let\@tempb\@empty
2331 \def\MT@temp{#1/#2/#3/#4/#5}%
2332 \MT@get@axis{encoding}{#1}%
2333 \MT@get@axis{family}{#2}%
2334 \MT@get@axis{series}{#3}%
2335 \MT@get@axis{shape}{#4}%

```

Append an asterisk for the size.

```

2336 \edef\@tempb{\@tempb*}%
2337 \MT@ifempty{#5}{%
2338 \MT@warn@axis@empty{size}{\string\normalsize}%
2339 \def\MT@val{*}%
2340 }{%
2341 \def\MT@val{#5}%
2342 }%
2343 \MT@get@size
2344 }%
2345 }

```

```

2346 \MT@define@code@key{encoding}{cfg}
2347 \MT@define@code@key{family}{cfg}
2348 \MT@define@code@key{series}{cfg}
2349 \MT@define@code@key{shape}{cfg}
2350 \MT@define@code@key@size{cfg}
2351 \MT@define@code@key@font{cfg}

```

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

```

2352 \def\MT@declare@codes#1{%
2353 \define@key{MT@#1@c}{name}[]{%
2354 \MT@ifempty{##1}\relax{\MT@def@n{MT@#1@c@name}{##1}}}%
2355 \define@key{MT@#1@c}{load}[]{%
2356 \MT@ifempty{##1}\relax{\def\MT@extra@load{##1}}}%
2357 \define@key{MT@#1@c}{factor}[]{%
2358 \MT@ifempty{##1}\relax{\def\MT@extra@factor{##1}}}%
2359 \define@key{MT@#1@c}{preset}[]{%
2360 \MT@ifempty{##1}\relax{\def\MT@extra@preset{##1}}}%

```

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

```

2361 \define@key{MT@#1@c}{context}[]{%
2362 \MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
2363 \define@key{MT@#1@c}{inputenc}{%

```



```
2364 \MT@ifempty{##1}\relax{\def\MT@extra@inputenc{##1}}%
2365 }
```

```
2366 \MT@map@clist@c\MT@features{\MT@declare@codes{#1}}
```

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

```
2367 \define@key{MT@pr@c}{unit}[character]{%
2368 \let\MT@extra@unit\@empty
2369 \KV@sp@def\@tempa{#1}%
2370 \MT@ifstreq\@tempa{relative}{%
2371 \MT@warning{Value `relative' for key `unit' is deprecated.\MessageBreak
2372 Use `unit=character' instead. For now, I'll do it.\MessageBreak
2373 for you}%
2374 \def\@tempa{character}%
2375 }\relax
2376 \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.

```
2377 \MT@ifdimen\@tempa{%
2378 \let\MT@extra@unit\@tempa
2379 }{%
2380 \MT@warning{\@tempa' is not a dimension.\MessageBreak
2381 Ignoring it and setting values relative to.\MessageBreak
2382 character widths}%
2383 }%
2384 }%
2385 }
```

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

```
2386 {*beta}
2387 \def\MT@define@key@unit#1{%
2388 \define@key{MT#1@c}{unit}[space]{%
2389 \let\MT@extra@unit\@empty
2390 \KV@sp@def\@tempa{##1}%
2391 \MT@ifstreq\@tempa{character}\relax{%
2392 \let\MT@extra@unit\m@ne
2393 \MT@ifstreq\@tempa{space}\relax{%
2394 \MT@ifdimen\@tempa{%
2395 \let\MT@extra@unit\@tempa
2396 }{%
2397 \MT@warning{\@tempa' is not a dimension.\MessageBreak
2398 Ignoring it and setting values relative to.\MessageBreak
2399 width of space}%
2400 }%
2401 }%
2402 }%
2403 }%
2404 }

2405 \MT@define@key@unit{sp}
2406 \MT@define@key@unit{kn}
2407 {/beta}
```

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

```
2408 \def\MT@define@ex@c@key#1{%
2409 \define@key{MT@ex@c}{#1}[]{%
2410 \MT@ifempty{##1}\relax{%
2411 \MT@ifint{##1}{%
```

A space terminates the number.

```
2412 \MT@def@n{MT@extra@#1}{##1 }%
2413 }%
```

```

2414     \MT@warning{%
2415         Value `##1' for option `#1' is not a number.\MessageBreak
2416         Ignoring it}%
2417     }%
2418 }%
2419 }%
2420 }

2421 \MT@define@ex@key{stretch}
2422 \MT@define@ex@key{shrink}
2423 \MT@define@ex@key{step}
2424 \define@key{MT@ex@key}{auto}[true]{%
2425     \KV@sp@def\@tempa{#1}%
2426     \csname if\@tempa\endcsname

```

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

```

2427     \MT@requires@pdftex4{%
2428         \def\MT@extra@auto{autoexpand}%
2429     }{%
2430         \MT@warning{pdfTeX too old for automatic font expansion}%
2431     }
2432 \else
2433     \MT@requires@pdftex4{%
2434         \let\MT@extra@auto\empty
2435     }\relax
2436 \fi
2437 }

```

### 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.

```

2438 \renewcommand*\DeclareCharacterInheritance[1][ ]{%
2439     \let\MT@extra@context\empty
2440     \let\MT@extra@inputenc\undefined
2441     \let\MT@inh@feat\empty
2442     \setkeys{MT@inh@}{#1}%
2443     \MT@begin@catcodes
2444     \MT@set@inh@list
2445 }

```

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

```

2446 \def\MT@set@inh@list#1#2{%
2447     \MT@ifempty\MT@inh@feat{%
2448         \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
2449     }{%
2450         \MT@map@clist@c\MT@inh@feat{{%
2451             \KV@sp@def\@tempa{#1}%
2452             \MT@ifempty\@tempa\relax{%
2453                 \MT@exp@one@n\MT@declare@char@inh
2454                 {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
2455             }%
2456         }}%
2457     }%
2458     \MT@end@catcodes
2459 }

```

The keys for the optional argument.

```
2460 \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
2461 \MT@map@clist@c\MT@features@long{%
2462   \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1}}
```

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

```
2463 \def\MT@declare@char@inh#1#2#3{%
2464   \MT@edef@n{MT@#1@inh@name}%
2465   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
2466   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
2467   \MT@ifdefined@cT\MT@extra@inputenc{%
2468     \global\MT@edef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
2469   <debug>\MT@info{1}{creating inheritance list ~\@nameuse{MT@#1@inh@name}'}%
2470   \global\MT@def@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
2471   \def\MT@permutelist{#1@inh}%
2472   \setkeys{MT@inh}{#2}%
2473   \MT@permute
2474 }
```

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

```
2475 \define@key{MT@inh}{encoding}[]{%
2476   \def\MT@val{#1}%
2477   \expandafter\MT@encoding@check\MT@val,\@nil
2478   \MT@get@highlevel{encoding}%
2479   \MT@edef@n{MT@tempencoding1}{\MT@val}%
2480 }
```

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

```
2481 \def\MT@encoding@check#1,#2\@nil{%
2482   \MT@ifempty{#2}\relax{%
2483     \edef\MT@val{#1}%
2484     \MT@warning{You may only specify one encoding for character\MessageBreak
2485               inheritance lists. Ignoring encoding(s) #2}%
2486   }%
2487 }
```

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

```
2488 \MT@define@code@key{family}{inh}
2489 \MT@define@code@key{series}{inh}
2490 \MT@define@code@key{shape}{inh}
2491 \MT@define@code@key@size{inh}
2492 \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`).

```
2493 \def\MT@inh@do#1,{%
2494   \ifx\relax#1\@empty \else
2495     \MT@inh@split #1==\relax
2496     \expandafter\MT@inh@do
2497   \fi
2498 }
```

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

```
2499 \def\MT@inh@split#1=#2=#3\relax{%
```

```

2500 \def\@tempa{#1}%
2501 \ifx\@tempa\@empty \else
2502   \MT@get@slot
2503   \ifnum\MT@char > \m@ne
2504     \let\MT@val\MT@char
2505     \MT@map@clist@n{#2}{%
2506       \def\@tempa{##1}%
2507       \ifx\@tempa\@empty \else
2508         \MT@get@slot
2509         \ifnum\MT@char > \m@ne
2510           \expandafter\MT@xadd
2511           \csname MT@inh@\MT@inh@name @\MT@val @\endcsname
2512           {\@tempa}%
2513         \fi
2514       \fi
2515     }%
2516 <*debug>
2517   \MT@dinfol{2}{children of #1 (\MT@val):
2518     \nameuse{MT@inh@\MT@inh@name @\MT@val @}}%
2519 </debug>
2520   \fi
2521 \fi
2522 }

```

### 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 `\MT@permute@` be the expansion of `\MT@<list type>@name`, i. e., the name of the currently defined list. `\MT@permute@@` Size ranges are held in a separate macro called `\MT@<list type>@/<font axes>@s@size`, `\MT@permute@@@` which in turn contains the respective `<list name>`s attached to the ranges.

```

2523 \def\MT@permute{%
2524   \let\MT@cnt@encoding\@ne
2525   \MT@permute@

```

Undefine commands for the next round.

```

2526   \MT@permute@reset
2527 }
2528 \def\MT@permute@{%
2529   \let\MT@cnt@family\@ne
2530   \MT@permute@@
2531   \MT@increment\MT@cnt@encoding
2532   \MT@ifdefined@nT{MT@tempencoding\MT@cnt@encoding}%
2533     \MT@permute@
2534 }
2535 \def\MT@permute@@{%
2536   \let\MT@cnt@series\@ne
2537   \MT@permute@@@
2538   \MT@increment\MT@cnt@family
2539   \MT@ifdefined@nT{MT@tempfamily\MT@cnt@family}%
2540     \MT@permute@@@
2541 }
2542 \def\MT@permute@@@{%
2543   \let\MT@cnt@shape\@ne
2544   \MT@permute@@@@
2545   \MT@increment\MT@cnt@series
2546   \MT@ifdefined@nT{MT@tempseries\MT@cnt@series}%
2547     \MT@permute@@@@
2548 }
2549 \def\MT@permute@@@@{%

```

```

2550 \MT@permute@@@@
2551 \MT@increment\MT@cnt@shape
2552 \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
2553 \MT@permute@@@@
2554 }

```

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

```

2555 \def\MT@permute@@@@{%
2556 \MT@permute@define{encoding}%
2557 \ifMT@document
2558 \ifx\MT@tempencoding\@empty \else
2559 \MT@ifdefined@n@TF{T\MT@tempencoding}\relax
2560 {\expandafter\expandafter\expandafter\@gobble}%
2561 \fi
2562 \fi
2563 \MT@permute@@@@
2564 }

```

\MT@permute@@@@

```

2565 \def\MT@permute@@@@{%
2566 \MT@permute@define{family}%
2567 \MT@permute@define{series}%
2568 \MT@permute@define{shape}%
2569 \edef\@tempa{\MT@tempencoding
2570 \MT@tempfamily
2571 \MT@tempseries
2572 \MT@tempshape
2573 \MT@ifdefined@c@T\MT@tempsize *}%

```

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

```

2574 \def\@tempb{////}%
2575 \ifx\@tempa\@tempb \else
2576 \ifx\MT@tempencoding\@empty
2577 \MT@warning{%
2578 You have to specify an encoding for\MessageBreak
2579 \nameuse{MT@abbr@\MT@permutelist} list
2580 ~\nameuse{MT@\MT@permutelist @name}'.\MessageBreak
2581 Ignoring it}%
2582 \else
2583 \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

2584 \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}{%
2585 \MT@map@tlist@c
2586 \MT@tempsize
2587 \MT@check@rlist
2588 }%
2589 \expandafter\MT@xaddb
2590 \csname MT@\MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
2591 \MT@tempsize
2592 *debug
2593 \MT@dinfo@n{1}{initializing: use list for font \@tempa,\MessageBreak
2594 sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
2595 @sizes\endcsname}%
2596 /debug
2597 }%

```

Only one list should apply to a given combination.

```

2598 \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context}{%
2599 \MT@warning{\nameuse{MT@abbr@\MT@permutelist} list

```

```

2600         \@nameuse{MT@MT@permutelist @name}' will override list\MessageBreak
2601         \@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
2602         for font \@tempa'}%
2603     }%
2604 *debug
2605     \MT@info@n1{1}{initializing: use list for font \@tempa
2606                 \ifx\MT@extra@context\@empty\else\MessageBreak
2607                 (context: \MT@extra@context)\fi}%
2608 /debug
2609     }%
2610     \global\MT@edef@n{MT@MT@permutelist @\@tempa\MT@extra@context}%
2611     {\csname MT@MT@permutelist @name\endcsname}%
2612     \fi
2613     \fi
2614 }

```

\MT@permute@define Define the commands.

```

2615 \def\MT@permute@define#1{%
2616   \expandafter\@tempcnta=\csname MT@cnt@#1\endcsname\relax
2617   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
2618   {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
2619   {\MT@let@nc{MT@temp#1}\@empty}%
2620 }

```

\MT@permute@reset Reset the commands.

```

2621 \def\MT@permute@reset{%
2622   \MT@permute@reset@{encoding}%
2623   \MT@permute@reset@{family}%
2624   \MT@permute@reset@{series}%
2625   \MT@permute@reset@{shape}%
2626   \let\MT@tempsize\undefined
2627 }

```

\MT@permute@reset@

```

2628 \def\MT@permute@reset@#1{%
2629   \@tempcnta=\@ne
2630   \MT@loop
2631   \MT@let@nc{MT@temp#1\the\@tempcnta}\undefined
2632   \advance\@tempcnta\@ne
2633   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
2634   \iftrue
2635   \iffalse
2636   \MT@repeat
2637 }

```

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

```

2638 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

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

```

2639 \def\MT@check@rlist@#1#2#3{%
2640   \def\@tempb{#1}%
2641   \def\@tempc{#2}%
2642   \@tempswafalse
2643   \expandafter\MT@map@tlist@
2644   \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
2645   \MT@check@range
2646 }

```

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

```

2647 \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.

```
2648 \def\MT@check@range@#1#2#3{%
2649   \MT@ifdim{#2}=\m@ne{%
2650     \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
2651       \MT@ifdim\@tempb={#1}\@tempswatruel\relax
2652     }{%
```

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

```
2653       \MT@ifdim\@tempb>{#1}\relax{%
2654         \MT@ifdim\@tempc>{#1}{%
2655           \@tempswatruel
2656           \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
2657         }\relax
2658       }%
2659     }%
2660   }{%
2661     \MT@ifdim\@tempc=\m@ne{%
```

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

```
2662       \MT@ifdim\@tempb<{#2}{%
2663         \MT@ifdim\@tempb<{#1}\relax\@tempswatruel
2664       }\relax
2665     }{%
```

- Both items are ranges.

```
2666       \MT@ifdim\@tempb<{#2}{%
2667         \MT@ifdim\@tempc>{#1}{%
2668           \@tempswatruel
2669           \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
2670         }\relax
2671       }\relax
2672     }%
2673   }%
2674   \if@tempswa
2675     \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
2676       ~\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
2677     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.

```
2678     \expandafter\MT@tlist@break
2679     \fi
2680   }
```

## 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 2681 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 2682 \newif\ifMT@opt@auto
2683 \newif\ifMT@opt@DVI
```

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

```

2684 \def\MT@define@option#1{%
2685   \define@key{MT}{#1}[true]{%
2686     \csname MT@opt@#1true\endcsname
2687     \MT@map@clist@n{##1}{%
2688       \KV@sp@def\MT@val{###1}%
2689       \MT@ifempty\MT@val\relax{%
2690         \csname MT@#1true\endcsname
2691         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2692         \MT@ifstreq\MT@val{true}\relax
2693         {%
2694           \MT@ifstreq\MT@val{false}{%
2695             \csname MT@#1false\endcsname
2696           }{%
2697             \MT@ifstreq\MT@val{compatibility}{%
2698               \MT@let@nc{MT@\@tempb @level}\@ne
2699             }{%
2700               \MT@ifstreq\MT@val{nocompatibility}{%
2701                 \MT@let@nc{MT@\@tempb @level}\tw@
2702               }{%

```

If everything failed, it should be a set name.

```

2703       \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
2704         \global\MT@edef@n{MT@\@tempb @setname}{\MT@val}%
2705       }{%
2706         \global\MT@edef@n{MT@\@tempb @setname}%
2707         {\@nameuse{MT@default@\@tempb @set}}%
2708         \MT@warning@n1{%
2709           The #1 set `~\MT@val' is undeclared.\MessageBreak
2710           Using set `~\@nameuse{MT@\@tempb @setname}' instead}%
2711       }%
2712     }%
2713   }%
2714 }%
2715 }%
2716 }%
2717 }%
2718 }%
2719 }
2720 \MT@define@option{protrusion}
2721 \MT@define@option{expansion}

```

`activate` is a shortcut for protrusion and expansion.

```

2722 \define@key{MT}{activate}[true]{%
2723   \setkeys{MT}{protrusion=#1}%
2724   \setkeys{MT}{expansion=#1}}%
2725 }
2726 <beta>

```

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

```

2727 \def\MT@define@option@#1{%
2728   \define@key{MT}{#1}[true]{%
2729     \csname MT@opt@#1true\endcsname
2730     \MT@map@clist@n{##1}{%
2731       \KV@sp@def\MT@val{###1}%
2732       \MT@ifempty\MT@val\relax{%
2733         \csname MT@#1true\endcsname
2734         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2735         \MT@ifstreq\MT@val{true}\relax
2736       }%

```



```

2737     \MT@ifstreq\MT@val{false}{%
2738     \csname MT@#1false\endcsname
2739     }{%
2740     \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
2741     \global\MT@edef@n{MT@\@tempb @setname}{\MT@val}%
2742     }{%
2743     \global\MT@edef@n{MT@\@tempb @setname}%
2744     {\@nameuse{MT@default@\@tempb @set}}%
2745     \MT@warning@n1{%
2746     The #1 set `~\MT@val' is undeclared.\MessageBreak
2747     Using set `~\@nameuse{MT@\@tempb @setname}' instead}%
2748     }%
2749     }%
2750     }%
2751     }%
2752     }%
2753     }%
2754 }

2755 \MT@define@option@{spacing}
2756 \MT@define@option@{kerning}
2757 </beta>

```

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

```

2758 \def\MT@def@bool@opt#1#2{%
2759 \define@key{MT}{#1}[]{%
2760 \MT@ifempty{#1}%
2761 {\def\@tempa{true}}%
2762 {\def\@tempa{#1}}%
2763 \MT@ifstreq\@tempa{true}\relax{%
2764 \MT@ifstreq\@tempa{false}\relax{%
2765 \MT@warning@n1{%
2766 ~##1' is not an admissible value for option\MessageBreak
2767 ~#1'. Assuming ~false'}%
2768 \def\@tempa{false}%
2769 }%
2770 }%
2771 #2%
2772 }%
2773 }

```

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

```

2774 \def\MT@def@simple@bool@opt#1{\MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
2775 \MT@map@tlist@n{{draft}{auto}{selected}}
2776 <beta> {babel}%
2777 }\MT@def@simple@bool@opt

```

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

```

2778 \MT@def@bool@opt{DVIoutput}{%
2779 \csname if\@tempa\endcsname
2780 \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
2781 \pdfoutput\z@
2782 \else
2783 \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
2784 \pdfoutput\@ne
2785 \fi
2786 }

```

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.

```
2787 \MT@def@bool@opt{defersetup}{%
2788   \csname if\@tempa\endcsname \else
2789     \AtEndOfPackage{%
2790       \MT@setup@
2791       \let\MT@setup@\@empty
2792       \let\MT@addto@setup\@firstofone
2793     }%
2794 \fi
2795 }
```

`final` is the opposite to `draft`.

```
2796 \MT@def@bool@opt{final}{%
2797   \csname if\@tempa\endcsname
2798   \MT@draftfalse
2799   \else
2800     \MT@drafttrue
2801 \fi
2802 }
```

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

```
2803 \define@key{MT}{verbose}[]{%
2804   \let\MT@vinfo\MT@info@nl
2805   \MT@ifempty{#1}%
2806   {\def\@tempa{true}}%
2807   {\def\@tempa{#1}}%
2808   \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
2809   \MT@ifstreq\@tempa{errors}{%
2810     \let\MT@warning\MT@warn@err
2811     \let\MT@warning@nl\MT@warn@err
2812   }{%
2813     \let\MT@vinfo@gobble
2814     \MT@ifstreq\@tempa{false}\relax{%
2815       \MT@warning@nl{%
2816         `#1' is not an admissible value for option\MessageBreak
2817         `verbose'. Assuming `false'}%
2818     }%
2819   }%
2820 }%
2821 }
```

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

```
2822 \def\MT@def@num@opt#1{%
2823   \define@key{MT}{#1}[]{%
2824     \MT@ifempty{#1}%
2825     {\MT@let@cn\@tempa{MT@#1@default}}%
2826     {\def\@tempa{#1 }}%
```

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

```
2827   \MT@ifint\@tempa{%
2828     \MT@edef@n{MT@#1}{\@tempa}%
2829   }{\MT@warning@nl{%
2830     Value `##1' for option `#1' is not a number.\MessageBreak
2831     Using default value of \number\@nameuse{MT@#1@default}}%
2832   }%
2833 }
```

```

2834 }
2835 \MT@map@tlist@n{{stretch}{shrink}{step}%
2836 beta {letterspacing}%
2837 }\MT@def@num@opt

```

factor will define the protrusion factor only.

```

2838 \define@key{MT}{factor}[]{}%
2839 \MT@ifempty{#1}%
2840   {\let\@tempa\MT@factor@default}%
2841   {\def\@tempa{#1}}%
2842 \MT@ifint\@tempa{%
2843   \edef\MT@pr@factor{\@tempa}%
2844 }{\MT@warning@nl{%
2845   Value `#1' for option `factor' is not a number.\MessageBreak
2846   Using default value of \number\MT@factor@default}%
2847 }%
2848 }

```

Unit for codes.

```

2849 \define@key{MT}{unit}[]{}%
2850 \MT@ifempty{#1}%
2851   {\def\@tempa{character}}%
2852   {\KV@@sp@def\@tempa{#1}}%
2853 \MT@ifstreq\@tempa{relative}{%
2854   \MT@warning{Value `relative' for option `unit' is deprecated.\MessageBreak
2855   Use `unit=character' instead. For now, I'll do it.\MessageBreak
2856   for you}%
2857   \def\@tempa{character}%
2858 } \relax
2859 \MT@ifstreq\@tempa{character} \relax{%
2860   \MT@ifdimen\@tempa{%
2861     \let\MT@pr@unit\@tempa
2862   }{%
2863     \MT@warning@nl{\@tempa' is not a dimension. Ignoring it and\MessageBreak
2864     setting values relative to character widths}%
2865   }%
2866 }%
2867 }

```

### 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  $\text{\TeX}$  systems are switching to the pdf $\text{\TeX}$  engine even for DVI output, so that the user might not even be aware of the fact that she's running pdf $\text{\TeX}$ .)

```

2868 \MT@protrusiontrue
2869 \ifnum\pdfoutput<\@ne \else

```

Also, we only enable expansion by default if pdf $\text{\TeX}$  can expand the fonts automatically.

```

2870 \MT@requires@pdftex4{
2871   \MT@expansiontrue
2872   \MT@autotrue
2873 } \relax
2874 \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.

```
2875 \define@key{MT}{config}[]{\relax}
2876 \def\MT@get@config#1config=#2,#3\@nil{%
2877   \MT@ifempty{#2}%
2878   {\def\MT@config@file{\MT@MT.cfg}}%
2879   {\KV@@sp@def\MT@config@file{#2.cfg}}%
2880 }
2881 \expandafter\expandafter\expandafter\MT@get@config
2882 \csname opt@\currname.\@current\endcsname,config=\@nil
```

Load the file.

```
2883 \IfFileExists{\MT@config@file}{%
2884   \MT@info@n1{Loading configuration file \MT@config@file}%
2885   \MT@begin@catcodes
2886   \let\MT@begin@catcodes\relax
2887   \let\MT@end@catcodes\relax
2888   \let\MT@curr@file\MT@config@file
2889   \input{\MT@config@file}%
2890   \endgroup
2891 }{\MT@warning@n1{%
2892   Could not find configuration file `~\MT@config@file'\!MessageBreak
2893   This will almost certainly cause undesired results.\MessageBreak
2894   Please fix your installation}%
2895 }
```

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

```
2896 \MT@map@clist@c\MT@features{%
2897   \MT@ifdefined@n@TF{MT@default@#1@set}\relax
2898   {\global\MT@def@n{MT@default@#1@set}{all}}%
2899 }
```

### 13.6.3 Hook for Other Packages

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

This hook was needed in versions prior to 1.9a to overcome the situation that (1) the `microtype` package should be loaded after all font defaults have been set up (hence, using `\ifpackageloaded` in the font package has not been viable), and (2) checking `\AtBeginDocument` could be too late, since fonts might already have been loaded, and consequently set up, in the preamble. With the new deferred setup, one could live without this command, however, it remains here since it’s simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

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

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

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

```
2900 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
2901 Command \string\MicroType@Hook\space is deprecated.\MessageBreak
2902 Use \string\MicroType@Hook\space instead}\MicroType@Hook}
2903 \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.

```
2904 \def\microtypesetup{\setkeys{MT}}
2905 \MT@addto@setup{\def\microtypesetup{\setkeys{MT}}}
2906 \def\MT@define@optionX#1#2{%
2907   \define@key{MTX}{#1}[true]{%
2908     \KV@sp@def\@tempb{#1}%
2909     \MT@map@clist@n{##1}{%
2910       \KV@sp@def\MT@val{###1}%
2911       \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
2912       \MT@ifempty\MT@val\relax{%
2913         \@tempcnta=\m@ne
2914         \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.

```
2915     \MT@checksetup\@tempb{%
2916       \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
2917       \MT@info{Enabling #1
2918         (level \number\csname MT@\@tempb @level\endcsname)}%
2919     }%
2920   }{%
2921     \MT@ifstreq\MT@val{false}{%
2922       \@tempcnta=\z@
2923       \MT@info{Disabling #1}%
2924     }{%
2925       \MT@ifstreq\MT@val{compatibility}{%
2926         \MT@checksetup\@tempb{%
2927           \@tempcnta=\@ne
2928           \MT@let@nc{MT@\@tempb @level}\@ne
2929           \MT@info{Setting #1 to level 1}%
2930         }%
2931       }{%
2932         \MT@ifstreq\MT@val{nocompatibility}{%
2933           \MT@checksetup\@tempb{%
2934             \@tempcnta=\tw@
2935             \MT@let@nc{MT@\@tempb @level}\tw@
2936             \MT@info{Setting #1 to level 2}%
2937           }%
2938         }{%
2939           \MT@warning{%
2940             Value `~\MT@val' for key `~#1' not recognized.\MessageBreak
2941             Use any of `true', `false', `compatibility' or\MessageBreak
2942             `nocompatibility'}%
2943           }%
2944         }%

```

```

2945     }%
2946   }%
2947   \ifnum\@tempcnta>\m@ne
2948     #2\@tempcnta\relax
2949   \fi
2950   }%
2951 }%
2952 }%
2953 }

```

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

```

2954 \def\MT@checksetup#1{%
2955   \expandafter\csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
2956   \expandafter\@firstofone
2957   \else
2958     \MT@warning{%
2959       You cannot enable \@nameuse{MT@abbr@#1} if it was disabled\MessageBreak
2960       in the package options,}%
2961     \expandafter\@gobble
2962   \fi
2963 }

2964 \MT@define@optionX{protrusion}\pdfprotrudechars
2965 \MT@define@optionX{expansion}\pdfadjustspacing
2966 <beta>

```

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

```

2967 \def\MT@define@optionX@#1#2{%
2968   \define@key{MTX}{#1}[true]{%
2969     \KV@sp@def\@tempb{#1}%
2970     \MT@map@clist@n{##1}{%
2971       \KV@sp@def\MT@val{###1}%
2972       \edef\@tempb{\csname MT@rbb@@\@tempb\endcsname}%
2973       \MT@ifempty\MT@val\relax{%
2974         \@tempcnta=\m@ne
2975         \MT@ifstreq\MT@val{true}{%
2976           \MT@checksetup\@tempb{%
2977             \@tempcnta=\@ne
2978             \MT@info{Enabling #1}%
2979           }%
2980         }%
2981         \MT@ifstreq\MT@val{false}{%
2982           \@tempcnta=\z@
2983           \MT@info{Disabling #1}%
2984         }%
2985         \MT@warning{%
2986           Value `~\MT@val' for key `~#1' not recognized.\MessageBreak
2987           Use either `true' or `false'~}%
2988       }%
2989     }%
2990     \ifnum\@tempcnta>\m@ne
2991       #2\relax
2992     \fi
2993   }%
2994 }%
2995 }%
2996 }

2997 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
2998 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
2999   \pdfappendkern \@tempcnta}
3000 <beta>
3001 \define@key{MTX}{activate}[true]{%

```

```

3002 \setkeys{MTX}{protrusion={#1}}%
3003 \setkeys{MTX}{expansion={#1}}%
3004 }

```

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

```

3005 \let\MT@saved@setupfont\MT@setupfont
3006 \define@key{MTX}{disable}[]{}%
3007 \MT@info{Inactivate `~\MT@MT' package}%
3008 \let\MT@setupfont\@gobble
3009 }
3010 \define@key{MTX}{enable}[]{}%
3011 \MT@info{Reactivate `~\MT@MT' package}%
3012 \let\MT@setupfont\MT@saved@setupfont
3013 }

```

### 13.6.5 Processing the Options

`\MT@ProcessOptionsWithKV` Parse options.

```

3014 \def\MT@ProcessOptionsWithKV#1{%
3015   \let\@tempc\relax
3016   \let\KV@tempa\@empty
3017   \MT@map@clist@c\@classoptionslist{%
3018     \def\CurrentOption{##1}%
3019     \MT@ifdefined@n@T{KV@#1@\CurrentOption}{%
3020       \edef\KV@tempa{\KV@tempa,\CurrentOption,}%
3021       \@expandtwoargs\@removeelement\CurrentOption
3022         \@unusedoptionlist\@unusedoptionlist
3023     }%
3024   }%
3025   \edef\KV@tempa{%
3026     \noexpand\setkeys{#1}{%
3027       \KV@tempa\@optionlist{\@currname.\@currentx}%
3028     }%
3029   }%
3030   \KV@tempa
3031   \AtEndOfPackage{\let\@unprocessedoptions\relax}%
3032   \let\CurrentOption\@empty
3033 }
3034 \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).

```

3035 \MT@addto@setup{%
3036 \ifMT@draft

```

We disable most of what we've just defined in the 3036 lines above if we are running in draft mode.

```

3037 \MT@warning@n1{`draft' option active.\MessageBreak
3038             Disabling all micro-typographic extensions.\MessageBreak
3039             This might lead to different line and page breaks}
3040 \MT@protrusionfalse
3041 \MT@expansionfalse
3042 *beta
3043 \MT@spacingfalse
3044 \MT@kerningfalse
3045 \MT@babelfalse
3046 /beta
3047 \let\MT@setupfont\relax

```

```

3048 \def\DeclareMicrotypeSet#1#\@gobbletwo}
3049 \renewcommand*UseMicrotypeSet[2] [] {}
3050 \renewcommand*SetProtrusion[3] [] {}
3051 \renewcommand*SetExpansion[3] [] {}
3052 *beta
3053 \renewcommand*SetExtraSpacing[3] [] {}
3054 \renewcommand*SetExtraKerning[3] [] {}
3055 /beta
3056 \renewcommand*DeclareCharacterInheritance[3] [] {}
3057 \renewcommand*DeclareMicrotypeAlias[2] {}
3058 \renewcommand*LoadMicrotypeFile[1] {}
3059 \renewcommand*microtypesetup[1] {}
3060 \renewcommand*microtypecontext[1] {}
3061 \else

```

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

```

3062 \ifnum\pdfoutput<\@ne
3063 \ifMT@opt@expansion \else
3064 \MT@expansionfalse
3065 \fi
3066 \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!

```

3067 \MT@info@n1{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3068 \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%

```

**Protrusion.**

```

3069 \ifMT@protrusion
3070 \edef\MT@active@features{\MT@active@features,pr}
3071 \pdfprotrudechars\MT@pr@level
3072 \MT@info@n1{Character protrusion enabled (level \number\MT@pr@level)%
3073 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3074 factor: \number\MT@pr@factor\fi
3075 \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`.

```

3076 \MT@ifdefined@c@TF\MT@pr@setname{%
3077 \MT@info@n1{Using protrusion set `~\MT@pr@setname'}%
3078 }{%
3079 \global\let\MT@pr@setname\MT@default@pr@set
3080 \MT@info@n1{Using default protrusion set `~\MT@pr@setname'}%
3081 }
3082 \else
3083 \let\MT@protrusion\relax
3084 \MT@info@n1{No character protrusion}
3085 \fi

```

**Expansion.**

```

3086 \ifMT@expansion

```



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

```
3087 \ifnum\MT@stretch=\m@ne
3088   \let\MT@stretch\MT@stretch@default
3089 \fi
```

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

```
3090 \ifnum\MT@shrink=\m@ne
3091   \ifnum\MT@stretch>\z@
3092     \let\MT@shrink\MT@stretch
3093   \else
3094     \let\MT@shrink\MT@shrink@default
3095   \fi
3096 \fi
```

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

```
3097 \ifnum\MT@step=\m@ne
3098   \ifnum\MT@stretch>\MT@shrink
3099     \ifnum\MT@shrink=\z@
3100       \@tempcnta=\MT@stretch
3101     \else
3102       \@tempcnta=\MT@shrink
3103     \fi
3104   \else
3105     \ifnum\MT@stretch=\z@
3106       \@tempcnta=\MT@shrink
3107     \else
3108       \@tempcnta=\MT@stretch
3109     \fi
3110   \fi
3111   \divide\@tempcnta 5\relax
3112 \else
3113   \@tempcnta=\MT@step
3114   \ifnum\@tempcnta=\z@
3115     \MT@warning@n1{The expansion step cannot be set to zero.\MessageBreak
3116     Setting it to one}
3117   \fi
3118 \fi
3119 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3120 \edef\MT@step{\number\@tempcnta\space}
```

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

```
3121 \let\MT@auto\@empty
3122 \ifMT@auto
3123   \MT@requires@pdfTeX4{%
```

We turn off automatic expansion if output mode is DVI.

```
3124   \ifnum\pdfoutput<\@ne
3125     \ifMT@opt@auto
3126       \MT@warning@n1{%
3127       Automatic font expansion only works for PDF output.\MessageBreak
3128       However, you are creating a DVI file. I will switch.\MessageBreak
3129       automatic font expansion off and hope that expanded.\MessageBreak
3130       fonts are available}
3131     \fi
3132     \MT@autofalse
3133   \else
3134     \def\MT@auto{autoexpand}
```

```
3135     \fi
```

Also, if pdfTeX is too old.

```
3136     }{%
3137     \ifMT@opt@auto
3138     \MT@warning@nl{%
3139     The pdftex you are using is too old for automatic\MessageBreak
3140     font expansion. I will switch it off and hope that\MessageBreak
3141     expanded fonts are available on your system.\MessageBreak
3142     Install pdftex version 1.20 or newer}
3143     \fi
3144     \MT@autofalse
3145     \def\MT@auto{1000 }
3146     }
```

No automatic expansion.

```
3147     \else
3148     \ifnum\MT@pdftex@no < 4
3149     \def\MT@auto{1000 }
3150     \fi
3151     \fi
```

Choose the appropriate macro for selected expansion.

```
3152     \ifMT@selected
3153     \let\MT@set@ex@codes\MT@set@ex@codes@s
3154     \else
3155     \let\MT@set@ex@codes\MT@set@ex@codes@n
3156     \fi
```

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

```
3157     \ifnum\MT@stretch=\z@
3158     \ifnum\MT@shrink=\z@
3159     \MT@warning@nl{%
3160     Both the stretch and shrink limit are set to zero.\MessageBreak
3161     Disabling font expansion}
3162     \MT@expansionfalse
3163     \fi
3164     \fi
3165     \fi
3166     \ifMT@expansion
3167     \edef\MT@active@features{\MT@active@features,ex}%
3168     \pdfadjustspacing\MT@ex@level
3169     \MT@info@nl{\ifMT@auto\else Non-\fi Automatic font expansion enabled
3170     (level \number\MT@ex@level),\MessageBreak
3171     stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3172     step: \number\MT@step, \ifMT@selected\else non-\fi selected}
3173     \MT@ifdefined@c@TF\MT@ex@setname{%
3174     \MT@info@nl{Using expansion set ~\MT@ex@setname'}%
3175     }{%
3176     \global\let\MT@ex@setname\MT@default@ex@set
3177     \MT@info@nl{Using default expansion set ~\MT@ex@setname'}%
3178     }
```

Inside \showhyphens, font expansion should be disabled.

```
3179     \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3180     \color@begingroup\everypar{}\parfillskip\z@skip
3181     \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3182     \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
```

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

```
3183     \gdef\showhyphens#1{\setbox0\vbox{%
3184     \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3185     \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
```

```

3186     \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}
3187 \else
3188   \let\MT@expansion\relax
3189   \MT@info@nl{No font expansion}
3190 \fi
3191 \fi
3192 }
3193 *beta
3194 \MT@requires@pdfTeX6{
3195   \MT@addto@setup{%

```

### Spacing.

```

3196   \ifMT@spacing
3197     \edef\MT@active@features{\MT@active@features,sp}
3198     \pdfadjustinterwordglue\@ne
3199     \MT@info@nl{Adjustment of interword spacing enabled}
3200     \MT@ifdefined@c@TF\MT@sp@setname{%
3201       \MT@info@nl{Using spacing set ~\MT@sp@setname'}%
3202     }{%
3203       \global\let\MT@sp@setname\MT@default@sp@set
3204       \MT@info@nl{Using default spacing set ~\MT@sp@setname'}%
3205     }
3206   \else
3207     \let\MT@spacing\relax
3208     \MT@info@nl{No adjustment of interword spacing}
3209   \fi

```

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

```

3210   \edef\MT@active@features{\MT@active@features,kn}
3211   \ifMT@kerning
3212     \pdfprependkern\@ne
3213     \pdfappendkern\@ne
3214     \MT@info@nl{Adjustment of character kerning enabled}
3215     \MT@ifdefined@c@TF\MT@kn@setname{%
3216       \MT@info@nl{Using kerning set ~\MT@kn@setname'}%
3217     }{%
3218       \global\let\MT@kn@setname\MT@default@kn@set
3219       \MT@info@nl{Using default kerning set ~\MT@kn@setname'}%
3220     }
3221   \else
3222     \MT@info@nl{No adjustment of character kerning}
3223   \fi
3224   \ifnum\MT@letterspacing=\m@ne
3225     \let\MT@letterspacing\MT@letterspacing@default
3226   \fi
3227 }

```

If pdfTeX is too old, we disable spacing and kerning.

```

3228 }{
3229   \MT@addto@setup{%
3230     \ifMT@spacing
3231       \MT@warning@nl{Adjustment of interword spacing only works with\MessageBreak
3232         pdfTeX version 1.40 or newer. Switching it off}%
3233     \else
3234       \MT@info@nl{No adjustment of interword spacing}
3235     \fi
3236     \MT@spacingfalse
3237     \let\MT@spacing\relax
3238     \ifMT@kerning
3239       \MT@warning@nl{Character kerning only works with\MessageBreak
3240         pdfTeX version 1.40 or newer. Switching it off}%

```

```

3241 \else
3242   \MT@info@nl{No adjustment of character kerning}
3243 \fi
3244 \MT@kerningfalse
3245 \let\MT@kerning\relax
3246 }
3247 }

```

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.

```

3248 \MT@requires@pdfTeX6{
3249 \AtBeginDocument{%
3250   \ifMT@spacing
3251     \ifMT@babel \else
3252       \ifnum\sfcode`. > 1500
3253         \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
3254           \MT@warning@nl{%
3255             \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3256             interword spacing will disable it. You might want\MessageBreak
3257             to add \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
3258             to your preamble}%
3259           }%
3260         \fi
3261       \fi
3262     \fi
3263   }
3264 } \relax
3265 /beta

```

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

```

3266 \MT@addto@setup{%
3267   \ifx\MT@active@features\@empty \else
3268     \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3269   \fi
3270   \MT@documenttrue
3271 }

```

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

```

3272 *beta
3273 \MT@addto@setup{%
3274   \ifMT@babel
3275     \@ifpackageloaded{babel}{%
3276       \MT@info@nl{Redefining babel's language switching commands}
3277       \let\MT@orig@select@language\select@language
3278       \def\select@language#1{%
3279         \MT@orig@select@language{#1}%
3280         \MT@ifdefined@n@TF{MT@babel@#1}{%
3281           \MT@vinfo{Changing to language `#1' on line \the\inputlineno}%
3282           \expandafter\MT@exp@one@n\expandafter\microtypecontext
3283           \csname MT@babel@#1\endcsname
3284         }{%
3285           \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3286         }%
3287       }
3288       \let\MT@orig@foreign@language\foreign@language
3289       \def\foreign@language#1{%

```

```

3290     \MT@orig@foreign@language{#1}%
3291     \MT@ifdefined@n@TF{MT@babel@#1}{%
3292         \MT@vinfo{Changing to context `#1' on line \the\inputlineno}%
3293         \expandafter\MT@exp@one@n\expandafter\microtypecontext
3294         \csname MT@babel@#1\endcsname
3295     }{%
3296         \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3297     }%
3298 }

```

Disable babel's active characters.

```

3299     \ifMT@kerning
3300     \@tempswafalse
3301     \ifpackagewith{babel}{french}\@tempswattrue\relax
3302     \ifpackagewith{babel}{frenchb}\@tempswattrue\relax
3303     \ifpackagewith{babel}{français}\@tempswattrue\relax
3304     \if@tempswa
3305     \NoAutoSpaceBeforeFDP
3306     \MT@warning@n1{Switching off French babel's active punctuation characters}%
3307     \fi
3308 \fi

```

In case babel was loaded before microtype:

```

3309     \MT@ifdefined@n@T{MT@babel@\language}{%
3310         \MT@vinfo{Changing to context `\'language' on line \the\inputlineno}%
3311         \expandafter\MT@exp@one@n\expandafter\microtypecontext
3312         \csname MT@babel@\language\endcsname
3313     }%
3314 }{%
3315     \MT@warning@n1{You did not load the babel package.\MessageBreak
3316         The `babel' option won't have any effect}
3317 }
3318 \fi
3319 }
3320 /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.

```

3321 \AtBeginDocument\selectfont

```

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

```

3322 \edef\MT@curr@file{\jobname.tex}

```

That was that.

```

3323 /package

```

## 14 Configuration Files

Let's now write the font configuration files.

```

3324 (*config)
3325

```

### 14.1 Font Sets

We first declare some sets in the main configuration file.

```

3326 (*m-t)
3327 %%% -----
3328 %%% FONT SETS

```

```

3329
3330 \DeclareMicrotypeSet{all}
3331 { }
3332
3333 \DeclareMicrotypeSet{allmath}
3334 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
3335
3336 \DeclareMicrotypeSet{alltext}
3337 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
3338
3339 \DeclareMicrotypeSet{basicmath}
3340 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
3341   family   = {rm*,sf*},
3342   series   = {md*},
3343   size     = {normalsize,footnotesize,small,large}
3344 }
3345
3346 \DeclareMicrotypeSet{basictext}
3347 { encoding = {OT1,T1,LY1,OT4,QX,T5},
3348   family   = {rm*,sf*},
3349   series   = {md*},
3350   size     = {normalsize,footnotesize,small,large}
3351 }
3352
3353 \DeclareMicrotypeSet{normalfont}
3354 { font = */*/*/*/* }
3355

```

The default sets.

```

3356 %% -----
3357 %% DEFAULT SETS
3358
3359 \DeclareMicrotypeSetDefault[protrusion]{alltext}
3360 \DeclareMicrotypeSetDefault[expansion]{basictext}
3361 {beta}
3362 \DeclareMicrotypeSetDefault[spacing]{basictext}
3363 \DeclareMicrotypeSetDefault[kerning]{alltext}
3364 {beta}
3365

```

## 14.2 Font Aliases

Fonts that are ‘the same’.

```

3366 %% -----
3367 %% FONT ALIASES
3368

```

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.

```

3369 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
3370 \DeclareMicrotypeAlias{aer}{cmr} % ae
3371 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
3372 \DeclareMicrotypeAlias{cmor}{cmr} % eco
3373 \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.

```

3374 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
3375 \DeclareMicrotypeAlias{qpl}{ppl} % qfonts/QuasiPalatino

```

```
3376 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
3377 \DeclareMicrotypeAlias{qtm}{ptm} % qfonts/QuasiTimes
```

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

The eulervm package virtually extends the Euler fonts.

```
3378 \DeclareMicrotypeAlias{zeur}{eur} % Euler VM
3379 \DeclareMicrotypeAlias{zeus}{eus}
```

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

```
3380 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
3381 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

Euro symbol fonts, to save some files.

```
3382 \DeclareMicrotypeAlias{zpeus}{zpeu} % Adobe Euro sans -> serif
3383 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
3384 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
3385
```

### 14.3 Interaction with babel

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

```
3386 <math>*\beta</math>
3387 %%% -----
3388 %%% INTERACTION WITH THE `babel' PACKAGE
3389
3390 \DeclareMicrotypeBabelHook
3391   {french,français}
3392   {kerning=french, spacing=}
3393
3394 \DeclareMicrotypeBabelHook
3395   {english,american,USenglish,british,UKenglish}
3396   {kerning=, spacing=nonfrench}
3397
3398 \DeclareMicrotypeBabelHook
3399   {turkish}
3400   {kerning=turkish, spacing=}
3401
3402 <math>/\beta</math>
```

### 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.

## 14.5 Character Inheritance

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

```
3403 </m-t>
3404 <*m-t|zpeu|mys>
3405 %%% -----
3406 %%% CHARACTER INHERITANCE
3407
3408 </m-t|zpeu|mys>
3409 <*m-t>
```

### 14.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 ('fi' ligature), 013 ('fl'), 014 ('ffi'), 015 ('ffl'),  $\text{Æ}$ ,  $\text{æ}$ ,  $\text{Œ}$ ,  $\text{œ}$ .

```
3410 \DeclareCharacterInheritance
3411 { encoding = OT1 }
3412 { f = {011}, % ff
3413   i = {\i},
3414   j = {\j},
3415   o = {\o},
3416   o = {\o},
3417 }
3418
```

### 14.5.2 T1

Candidates here: 028 ('fi'), 029 ('fl'), 030 ('ffi'), 031 ('ffl'), 156 ('IJ' ligature, since  $\LaTeX$  2005/12/01 accessible as `\IJ`), 188 ('ij', `\ij`),  $\text{Æ}$ ,  $\text{æ}$ ,  $\text{Œ}$ ,  $\text{œ}$ .

```
3419 \DeclareCharacterInheritance
3420 { encoding = T1 }
3421 { A = {\^A,\'A,\^A,\-A,\"A,\r A,\k A,\u A},
3422   a = {\^a,\'a,\^a,\-a,\"a,\r a,\k a,\u a},
3423   C = {\'C,\c C,\v C},
3424   c = {\'c,\c c,\v c},
3425   D = {\v D,\DH},
3426   d = {\v d,\dj},
3427   E = {\^E,\'E,\^E,\"E,\k E,\v E},
3428   e = {\^e,\'e,\^e,\"e,\k e,\v e},
3429   f = {027}, % ff
3430   G = {\u G},
3431   g = {\u g},
3432   I = {\^I,\'I,\^I,\"I,\.I},
3433   i = {\^i,\'i,\^i,\"i,\i},
```



```

3434     j = {\j},
3435     L = {\L,\'L,\v L},
3436     l = {\l,\'l,\v l},
3437     N = {\'N,\-N,\v N},
3438     n = {\'n,\-n,\v n},
3439     O = {\0,\^0,\'0,\^0,\-0,\"0,\H 0},
3440     o = {\o,\^o,\'o,\^o,\-o,\"o,\H o},
3441     R = {\'R,\v R},
3442     r = {\'r,\v r},
3443     S = {\'S,\c S,\v S,\SS},
3444     s = {\'s,\c s,\v s},
3445     T = {\c T,\v T},
3446     t = {\c t,\v t},
3447     U = {\~U,\'U,\^U,\"U,\H U,\r U},
3448     u = {\~u,\'u,\^u,\"u,\H u,\r u},
3449     Y = {\'Y,\"Y},
3450     y = {\'y,\"y},
3451     Z = {\'Z,\.Z,\v Z},
3452     z = {\'z,\.z,\v z},
3453     - = {127},
3454 }
3455

```

### 14.5.3 LY1

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

```

3456 \DeclareCharacterInheritance
3457 { encoding = LY1 }
3458 { A = {\^A,\'A,\^A,\-A,\"A,\r A},
3459   a = {\~a,\'a,\^a,\-a,\"a,\r a},
3460   C = {\c C},
3461   c = {\c c},
3462   D = {\DH},
3463   E = {\~E,\'E,\^E,\"E},
3464   e = {\~e,\'e,\^e,\"e},
3465   f = {011}, % ff
3466   I = {\~I,\'I,\^I,\"I},
3467   i = {\~i,\'i,\^i,\"i,\i},
3468   L = {\L},
3469   l = {\l},
3470   N = {\-N},
3471   n = {\-n},
3472   O = {\~O,\'O,\^O,\-O,\"O,\O},
3473   o = {\~o,\'o,\^o,\-o,\"o,\o},
3474   S = {\v S},
3475   s = {\v s},
3476   U = {\~U,\'U,\^U,\"U},
3477   u = {\~u,\'u,\^u,\"u},
3478   Y = {\'Y,\"Y},
3479   y = {\'y,\"y},
3480   Z = {\v Z},
3481   z = {\v z},
3482 }
3483

```

### 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.

```

3484 \DeclareCharacterInheritance

```

```

3485 { encoding = OT4 }
3486 { A = {\k A},
3487   a = {\k a},
3488   C = {\'C},
3489   c = {\'c},
3490   E = {\k E},
3491   e = {\k e},
3492   f = {011}, % ff
3493   i = {\i},
3494   j = {\j},
3495   L = {\L},
3496   l = {\l},
3497   N = {\'N},
3498   n = {\'n},
3499   O = {\0,\'0},
3500   o = {\o,\'o},
3501   S = {\'S},
3502   s = {\'s},
3503   Z = {\'Z,\.Z},
3504   z = {\'z,\.z},
3505 }
3506

```

#### 14.5.5 QX

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

```

3507 \DeclareCharacterInheritance
3508 { encoding = QX }
3509 { A = {\^A,\'A,\^A,\~A,\"A,\k A,\AA},
3510   a = {\^a,\'a,\^a,\~a,\"a,\k a,\aa},
3511   C = {\'C,\c C},
3512   c = {\'c,\c c},
3513   D = {\DH},
3514   E = {\^E,\'E,\^E,\"E,\k E},
3515   e = {\^e,\'e,\^e,\"e,\k e},
3516   f = {011}, % ff
3517   I = {\^I,\'I,\^I,\"I,\k I},
3518   i = {\^i,\'i,\^i,\"i,\k i,\i},
3519   j = {\j},
3520   L = {\L},
3521   l = {\l},
3522   N = {\'N,\~N},
3523   n = {\'n,\~n},
3524   O = {\0,\^0,\'0,\^0,\~0,\"0},
3525   o = {\o,\^o,\'o,\^o,\~o,\"o},
3526   S = {\'S,\c S,\v S},
3527   s = {\'s,\c s,\v s},
3528   T = {\c T},
3529   t = {\c t},
3530   U = {\^U,\'U,\^U,\"U,\k U},
3531   u = {\^u,\'u,\^u,\"u,\k u},
3532   Y = {\'Y,\"Y},
3533   y = {\'y,\"y},
3534   Z = {\'Z,\.Z,\v Z},
3535   z = {\'z,\.z,\v z},
3536   . = \textellipsis
3537 }
3538

```

### 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.

```

3539 \DeclareCharacterInheritance
3540   { encoding = T5 }
3541   { A = {\^A,\'A,\-A,\h A,\d A,\^A,\u A,
3542         \~\Acircumflex,\'\Acircumflex,\-\Acircumflex,\h\Acircumflex,\d\Acircumflex,
3543         \~\Abreve,\'\Abreve,\-\Abreve,\h\Abreve,\d\Abreve},
3544     a = {\^a,\'a,\-a,\h a,\d a,\^a,\u a,
3545         \~\acircumflex,\'\acircumflex,\-\acircumflex,\h\acircumflex,\d\acircumflex,
3546         \~\abreve,\'\abreve,\-\abreve,\h\abreve,\d\abreve},
3547     D = {DJ},
3548     d = {dj},
3549     E = {\^E,\'E,\-E,\h E,\d E,\^E,
3550         \~\Ecircumflex,\'\Ecircumflex,\-\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
3551     e = {\^e,\'e,\-e,\h e,\d e,\^e,
3552         \~\ecircumflex,\'\ecircumflex,\-\ecircumflex,\h\ecircumflex,\d\ecircumflex},
3553     I = {\^I,\'I,\-I,\h I,\d I},
3554     i = {\^i,\'i,\-i,\h i,\d i,\i},
3555     O = {\^O,\'O,\-O,\h O,\d O,\^O,\horn O,
3556         \~\Ocircumflex,\'\Ocircumflex,\-\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
3557         \~\Ohorn,\'\Ohorn,\-\Ohorn,\h\Ohorn,\d\Ohorn},
3558     o = {\^o,\'o,\-o,\h o,\d o,\^o,\horn o,
3559         \~\ocircumflex,\'\ocircumflex,\-\ocircumflex,\h\ocircumflex,\d\ocircumflex,
3560         \~\ohorn,\'\ohorn,\-\ohorn,\h\ohorn,\d\ohorn},
3561     U = {\^U,\'U,\-U,\h U,\d U,\horn U,
3562         \~\Uhorn,\'\Uhorn,\-\Uhorn,\h\Uhorn,\d\Uhorn},
3563     u = {\^u,\'u,\-u,\h u,\d u,\horn u,
3564         \~\uhorn,\'\uhorn,\-\uhorn,\h\uhorn,\d\uhorn},
3565     Y = {\^Y,\'Y,\-Y,\h Y,\d Y},
3566     y = {\^y,\'y,\-y,\h y,\d y},
3567   }
3568
3569 /m-t

```

### 14.5.7 Euro symbols

Make Euro symbols settings simpler.

```

3570 <zpeu>
3571 \DeclareCharacterInheritance
3572   { encoding = U,
3573     family = {zpeu,zpeus,eurosans} }
3574   { E = 128 }
3575
3576 </zpeu>
3577 <mvs>
3578 \DeclareCharacterInheritance
3579   { encoding = OT1,
3580     family = mvs }
3581   { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
3582

```

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.

```

3583 \DeclareCharacterInheritance
3584   { encoding = U,
3585     family = mvs }
3586   { 164 = {099,100,101} }

```

```
3587
3588 </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).

```
3589 <*m-t>
3590 %%% -----
3591 %%% EXPANSION SETTINGS
3592
3593 \SetExpansion
3594 [ name = default ]
3595 { encoding = {OT1,OT4,QX,T1,LY1} }
3596 {
3597     A = 500,    a = 700,
3598     \AE = 500, \ae = 700,
3599     B = 700,    b = 700,
3600     C = 700,    c = 700,
3601     D = 500,    d = 700,
3602     E = 700,    e = 700,
3603     F = 700,
3604     G = 500,    g = 700,
3605     H = 700,    h = 700,
3606     K = 700,    k = 700,
3607     M = 700,    m = 700,
3608     N = 700,    n = 700,
3609     O = 500,    o = 700,
3610     \OE = 500, \oe = 700,
3611     P = 700,    p = 700,
3612     Q = 500,    q = 700,
3613     R = 700,
3614     S = 700,    s = 700,
3615     U = 700,    u = 700,
3616     W = 700,    w = 700,
3617     Z = 700,    z = 700,
3618     2 = 700,
3619     3 = 700,
3620     6 = 700,
3621     8 = 700,
3622     9 = 700,
3623 }
3624
```

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

```
3625 \SetExpansion
3626 [ name = T5 ]
3627 { encoding = T5 }
3628 {
3629     A = 500,    a = 700,
3630     B = 700,    b = 700,
3631     C = 700,    c = 700,
3632     D = 500,    d = 700,
3633     E = 700,    e = 700,
3634     F = 700,
3635     G = 500,    g = 700,
3636     H = 700,    h = 700,
3637     K = 700,    k = 700,
3638     M = 700,    m = 700,
3639     N = 700,    n = 700,
3640     O = 500,    o = 700,
```

```

3641 P = 700,    p = 700,
3642 Q = 500,    q = 700,
3643 R = 700,
3644 S = 700,    s = 700,
3645 U = 700,    u = 700,
3646 W = 700,    w = 700,
3647 Z = 700,    z = 700,
3648 2 = 700,
3649 3 = 700,
3650 6 = 700,
3651 8 = 700,
3652 9 = 700,
3653 }
3654
3655 </m-t>

```

## 14.7 Character Protrusion

```

3656 %%% -----
3657 %%% PROTRUSION SETTINGS
3658

```

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.

```

3659 <*cfg-t>
3660 \SetProtrusion
3661 <m-t> [ name = default ]

```

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

```

3662 <bch> [ name      = bch-default ]
      Computer Modern Roman (cmr),
3663 <cmr> [ name      = cmr-default ]
      Adobe Garamond (pad, padx, padj),
3664 <pad> [ name      = pad-default ]
      Minion16 (pmnx, pmnj),
3665 <pmn> [ name      = pmnj-default ]
      Palatino (ppl, pplx, pplj),
3666 <ppl> [ name      = ppl-default ]
      Times (ptm, ptmx, ptmj),
3667 <ptm> [ name      = ptm-default ]
      and URW Garamond (ugm).
3668 <ugm> [ name      = ugm-default ]
3669 <m-t> { encoding = OT1      }
3670 <cmr> { }
3671 <bch|pad|pmn|ugm> { encoding = OT1,
3672 <ppl|ptm> { encoding = {OT1,OT4},
3673 <bch>      family = bch }
3674 <pad>      family = {pad,padx,padj} }
3675 <pmn>      family = pmnj }
3676 <ppl>      family = {ppl,pplx,pplj} }
3677 <ptm>      family = {ptm,ptmx,ptmj} }
3678 <ugm>      family = ugm }
3679 {
3680 <m-t|bch|cmr|pad|pmn|ppl|ptm>    A = {50,50},
3681 <ugm>      A = {50,100},
3682 <m-t|pad|ptm> \AE = {50, },
3683 <ugm>      \AE = {150,50},
3684 <ugm>      B = { ,50},
3685 <bch|pad|pmn|ugm>    C = {50, },
3686 <bch|pad|pmn>      D = { ,50},
3687 <ugm>      D = { ,70},
3688 <ugm>      E = { ,50},
3689 <m-t|bch|cmr|pad|pmn|ptm>    F = { ,50},
3690 <ugm>      F = { ,70},
3691 <bch|pad|pmn>      G = {50, },
3692 <ugm>      G = {50,50},
3693 <m-t|cmr|pad|pmn|ppl|ptm|ugm>  J = {50, },
3694 <bch>      J = {100, },
3695      K = { ,50},
3696 <m-t|bch|cmr|pad|pmn|ppl>    L = { ,50},
3697 <ptm>      L = { ,80},
3698 <ugm>      L = { ,120},
3699 <bch|pad|pmn|ugm>    O = {50,50},
3700 <pad|pmn> \OE = {50, },
3701 <ugm>      \OE = {50,50},
3702 <ugm>      P = { ,50},
3703 <bch|pad|pmn>      Q = {50,70},
3704 <ugm>      Q = {50,50},
3705 <bch>      R = { ,50},
3706 <ugm>      R = { ,70},
3707 <m-t|bch|cmr|pad|pmn|ppl|ptm> T = {50,50},
3708 <ugm>      T = {70,70},
3709 <m-t|bch|cmr|pad|pmn|ppl|ptm> V = {50,50},

```

16 Contributed by Harald Harders (h.harders@tu-bs.de)

```

3710 <ugm>      V = {70,70},
3711 <m-t|bch|cmr|pad|pmn|ppl|ptm>  W = {50,50},
3712 <ugm>      W = {70,70},
3713 <m-t|bch|cmr|pad|pmn|ppl|ptm>  X = {50,50},
3714 <ugm>      X = {50,70},
3715 <m-t|bch|cmr|pad|pmn|ppl>      Y = {50,50},
3716 <ptm|ugm>  Y = {80,80},
3717 <ugm>      Z = {50,50},
3718 <m-t|bch|cmr|pad|pmn|ppl|ptm>  k = { ,50},
3719 <ugm>      k = { ,70},
3720 <pmn>      l = { , -50},
3721 <pad|ppl>   p = {50,50},
3722 <ugm>      p = { ,50},
3723 <pad|ppl>   q = {50, },
3724          r = { ,50},
3725 <cmr|pad|pmn>  t = { ,70},
3726 <bch>       t = { ,50},
3727 <ugm>      t = { ,100},
3728 <m-t|bch|cmr|pad|pmn|ppl|ptm>  v = {50,50},
3729 <ugm>      v = {50,70},
3730 <m-t|bch|cmr|pad|pmn|ppl|ptm>  w = {50,50},
3731 <ugm>      w = {50,70},
3732          x = {50,50},
3733 <m-t|bch|pad|pmn>  y = { ,50},
3734 <cmr|ppl|ptm>    y = {50,70},
3735 <ugm>          y = { ,70},

3736 <cmr>       0 = { ,50},
3737 <m-t>       1 = {50,50},
3738 <bch|pad|ptm|ugm> 1 = {150,150},
3739 <cmr>       1 = {100,200},
3740 <pmn>       1 = { ,50},
3741 <ppl>       1 = {100,100},
3742 <bch|cmr|pad|ugm> 2 = {50,50},
3743 <cmr|pad|ugm> 3 = {50,50},
3744 <bch|pmn>   3 = {50, },
3745 <m-t|pad>   4 = {50,50},
3746 <bch>       4 = {100,50},
3747 <cmr|ugm>  4 = {70,70},
3748 <pmn>       4 = {50, },
3749 <ptm>       4 = {70, },
3750 <cmr>       5 = { ,50},
3751 <pad>       5 = {50,50},
3752 <bch>       6 = {50, },
3753 <cmr>       6 = { ,50},
3754 <pad>       6 = {50,50},
3755 <m-t>       7 = {50,50},
3756 <bch|pad|pmn|ugm> 7 = {50,80},
3757 <cmr|ptm>   7 = {50,100},
3758 <ppl>       7 = { ,50},
3759 <cmr>       8 = { ,50},
3760 <bch|pad>   9 = {50,50},
3761 <cmr>       9 = { ,50},
3762 <m-t|cmr|pad|pmn|ppl|ptm|ugm> . = { ,700},
3763 <bch>       . = { ,600},
3764          {,} = { ,500},
3765 <m-t|cmr|pad|pmn|ppl|ptm|ugm> : = { ,500},
3766 <bch>       : = { ,400},
3767 <m-t|bch|pad|pmn|ptm> ; = { ,300},
3768 <cmr|ppl>   ; = { ,500},
3769 <ugm>       ; = { ,400},
3770          ! = { ,100},
3771 <m-t|pad|pmn|ptm> ? = { ,100},

```

```

3772 <bch|cmr|ppl|ugm>    ? = { ,200},
3773 <pmn>                " = {300,300},
3774 <m-t|bch|cmr|pad|pmn|ppl>    @ = {50,50},
3775 <ptm>                @ = {100,100},
3776 <m-t|bch|cmr|pad|pmn|ppl|ptm>    ~ = {200,250},
3777 <ugm>                ~ = {300,350},
3778 <pad|ppl|ptm>        & = {50,100},
3779 <ugm>                & = { ,100},
3780 <m-t|cmr|pad|pmn>    \% = {50,50},
3781 <bch>                \% = { ,50},
3782 <ppl|ptm>           \% = {100,100},
3783 <ugm>                \% = {50,100},
3784 <m-t|ppl|ptm|ugm>    * = {200,200},
3785 <bch|pmn>            * = {200,300},
3786 <cmr|pad>            * = {300,300},
3787 <m-t|cmr|ppl|ptm>    + = {250,250},
3788 <bch>                + = {150,250},
3789 <pad>                + = {300,300},
3790 <pmn>                + = {150,200},
3791 <ugm>                + = {250,300},
3792 <ugm>                {=} = {200,200},
3793 <m-t|pad|pmn|ptm>    ( = {100, }, ) = { ,200},
3794 <bch|ugm>            ( = {200, }, ) = { ,200},
3795 <cmr|ppl>            ( = {100, }, ) = { ,300},
3796 <bch|pmn>            [ = {100, }, ] = { ,100},

3797 <m-t|pad|pmn|ptm>    / = {100,200},
3798 <bch>                / = { ,200},
3799 <cmr|ppl>            / = {200,300},
3800 <ugm>                / = {100,300},
3801 <m-t|ptm>            - = {500,500},
3802 <bch|cmr|ppl>        - = {400,500},
3803 <pad>                - = {300,500},
3804 <pmn>                - = {200,400},
3805 <ugm>                - = {500,600},
3806 <m-t|pmn>            \textendash = {200,200}, \textendash = {150,150},
3807 <bch>                \textendash = {200,300}, \textendash = {150,250},
3808 <cmr>                \textendash = {400,300}, \textendash = {300,200},
3809 <pad|ppl|ptm>        \textendash = {300,300}, \textendash = {200,200},
3810 <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).

```

3811 <m-t|bch|pmn>        \textquoteleft = {300,400}, \textquoteright = {300,400},
3812 <cmr>                \textquoteleft = {500,700}, \textquoteright = {500,600},
3813 <pad|ppl>            \textquoteleft = {500,700}, \textquoteright = {500,700},
3814 <ptm>                \textquoteleft = {500,500}, \textquoteright = {300,500},
3815 <ugm>                \textquoteleft = {300,600}, \textquoteright = {300,600},
3816 <m-t|bch|pmn>        \textquotedblleft = {300,300}, \textquotedblright = {300,300},
3817 <cmr>                \textquotedblleft = {500,300}, \textquotedblright = {200,600},
3818 <pad|ppl|ptm>        \textquotedblleft = {300,400}, \textquotedblright = {300,400},
3819 <ugm>                \textquotedblleft = {400,400}, \textquotedblright = {400,400},
3820 }
3821

```

Greek uppercase letters are in OT1 encoding only.

```

3822 <*cmr>
3823 \SetProtrusion
3824 [ name = cmr-OT1,
3825 load = cmr-default ]
3826 { encoding = {OT1,OT4},
3827 family = cmr }

```



```

3828 {
3829   \AE = { 50, },
3830   "00 = { ,150}, % \Gamma
3831   "01 = {100,100}, % \Delta
3832   "02 = { 50, 50}, % \Theta
3833   "03 = {100,100}, % \Lambda
3834   "06 = { 50, 50}, % \Sigma
3835   "07 = {100,100}, % \Upsilon
3836   "08 = { 50, 50}, % \Phi
3837   "09 = { 50, 50}, % \Psi

```

Remaining slots can be found in the source file.

```

3838 }
3839
3840 </cmr>

```

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

```

3841 \SetProtrusion
3842 <m-t> [ name = T1-default,
3843 <bch> [ name = bch-T1,
3844 <cmr> [ name = cmr-T1,
3845 <pad> [ name = pad-T1,
3846 <pmn> [ name = pmnj-T1,
3847 <ppl> [ name = ppl-T1,
3848 <ptm> [ name = ptm-T1,
3849 <ugm> [ name = ugm-T1,
3850 <m-t> load = default ]
3851 <bch> load = bch-default ]
3852 <cmr> load = cmr-default ]
3853 <pad> load = pad-default ]
3854 <pmn> load = pmnj-default ]
3855 <ppl> load = ppl-default ]
3856 <ptm> load = ptm-default ]
3857 <ugm> load = ugm-default ]
3858 <m-t> { encoding = {T1,LY1} }
3859 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
3860 <ptm|ugm> { encoding = {T1},
3861 <bch> family = bch }
3862 <cmr> family = cmr }
3863 <pad> family = {pad,padx,padj} }
3864 <pmn> family = pmnj }
3865 <ppl> family = {ppl,pplx,pplj} }
3866 <ptm> family = {ptm,ptmx,ptmj} }
3867 <ugm> family = ugm }
3868 {
3869 <cmr> \AE = {50, },
3870 <bch> \OE = {50, },
3871 <pmn> \TH = { ,50},
3872 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
3873 <cmr> _ = {200,200},
3874 <ugm> _ = {100,200},
3875 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
3876 <bch> \textbackslash = {150,200},
3877 <cmr|ppl> \textbackslash = {200,300},
3878 <ugm> \textbackslash = {100,300},
3879 <ugm> \textbar = {200,200},
3880 <cmr> \textquotedblleft = {200,600},
3881 <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.

```

3882 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3883 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
3884 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
3885 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
3886 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
3887 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
3888 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
3889 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
3890 <pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
3891 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
3892 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
3893 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
3894 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
3895 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
3896 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200},
3897 <pmn> \textless = {100, }, \textgreater = { ,100},
3898 <pmn> \textvisiblespace = {100,100}, % not in LY1
3899 }
3900

```

The lmodern fonts, on the other hand, restore the original kerning from the OT1 fonts, and so do we. Silly, isn't it?

```

3901 <*cmr>
3902 \SetProtrusion
3903 [ name = lmr-T1,
3904 load = cmr-T1 ]
3905 { encoding = {T1,LY1},
3906 family = lmr }
3907 {
3908 \textquotedblleft = {500,300},
3909 \quotedblbase = {500,300},
3910 }
3911
3912 </cmr>

```

Settings for the QX encoding (generic and Times). It also includes some glyphs otherwise in TS1.

```

3913 <*m-t|ptm>
3914 \SetProtrusion
3915 <m-t> [ name = QX-default,
3916 <ptm> [ name = ptm-QX,
3917 <m-t> load = default ]
3918 <ptm> load = ptm-default ]
3919 <m-t> { encoding = QX }
3920 <ptm> { encoding = QX,
3921 <ptm> family = {ptm,ptmx,ptmj} }
3922 {
3923 <ptm> * = {200,200},
3924 {=} = {100,100},
3925 \textunderscore = {100,100},
3926 \textbackslash = {100,200},
3927 \quotedblbase = {400,400},
3928 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
3929 <ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
3930 \textexclamdown = {100, }, \textquestiondown = {100, },
3931 <m-t> \textbraceleft = {400,200}, \textbraceright = {200,400},
3932 <ptm> \textbraceleft = {200,200}, \textbraceright = {200,300},
3933 \textless = {200,100}, \textgreater = {100,200},
3934 \textminus = {200,200}, \textdegree = {300,300},
3935 <m-t> \copyright = {100,100}, \textregistered = {100,100},
3936 <ptm> \copyright = {100,150}, \textregistered = {100,150},

```

```

3937 <ptm> \textxgeq = { ,100}, \textxleq = {100, },
3938 <ptm> \textalpha = { , 50}, \textDelta = { 70, 70},
3939 <ptm> \textpi = { 50, 80}, \textSigma = { , 70},
3940 <ptm> \textmu = { , 80}, \texteuro = { 50, 50},
3941 <ptm> \textellipsis = {150,200}, \textasciitilde = { 80, 80},
3942 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
3943 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
3944 <ptm> \textdiv = { 50,150}, \textsection = { 80, 80},
3945 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
3946 <ptm> \textbullet = {150,150}, \textperiodcentered = {300,300},
3947 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
3948 <ptm> \textperthousand = { ,50},
3949 }
3950
3951 </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.

```

3952 <*cmr|bch>
3953 \SetProtrusion
3954 <cmr> [ name = cmr-T5,
3955 <cmr> load = cmr-default ]
3956 <bch> [ name = bch-T5,
3957 <bch> load = bch-default ]
3958 { encoding = T5,
3959 <cmr> family = cmr }
3960 <bch> family = bch }
3961 {
3962 <bch> _ = {100,100},
3963 <bch> \textbackslash = {150,200},
3964 <cmr> \textbackslash = {200,300},
3965 <cmr> \textquotedblleft = {200,600},
3966 <cmr> \textquotedbl = {300,300},
3967 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
3968 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3969 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
3970 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
3971 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},
3972 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
3973 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
3974 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
3975 \textless = {200,100}, \textgreater = {100,200},
3976 }
3977
3978 </cmr|bch>
3979 <*pmn>
3980 \SetProtrusion
3981 [ name = pmnx-OT1,
3982 load = pmnj-default ]
3983 { encoding = OT1,
3984 family = pmnx }
3985 {
3986 1 = {230,180},
3987 }
3988
3989 \SetProtrusion
3990 [ name = pmnx-T1,
3991 load = pmnj-T1 ]
3992 { encoding = {T1,LY1},
3993 family = pmnx }
3994 {
3995 1 = {230,180},

```

```

3996 }
3997
3998 </pmn>

```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```

3999 <*ptm>
4000 \SetProtrusion
4001 [ name = ptm-LY1,
4002   load = ptm-T1 ]
4003 { encoding = LY1,
4004   family = {ptm,ptmx,ptmj} }
4005 {
4006   _ = {100,100},
4007   \texttrademark = {100,100},
4008   \textregistered = {100,100},
4009   \textcopyright = {100,100},
4010   \textdegree = {300,300},
4011   \textminus = {200,200},
4012   \textellipsis = {150,200},
4013   \texteuro = { , }, % ?
4014   \textcent = {100,100},
4015   \textquotesingle = {500,500},
4016   \textflorin = { 50, 70},
4017   \textdagger = {150,150},
4018   \textdaggerdbl = {100,100},
4019   \textperthousand = { , 50},
4020   \textbullet = {150,150},
4021   \textonesuperior = {100,100},
4022   \texttwosuperior = { 50, 50},
4023   \textthreesuperior = { 50, 50},
4024   \textperiodcentered = {300,300},
4025   \textplusminus = { 50, 80},
4026   \textmultiply = {100,100},
4027   \textdivide = { 50,150},

```

Remaining slots in the source file.

```

4028 }
4029
4030 </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.

```

4031 \SetProtrusion
4032 <m-t> [ name = OT1-it ]
4033 <bch> [ name = bch-it ]
4034 <cmr> [ name = cmr-it ]
4035 <pad> [ name = pad-it ]
4036 <pmn> [ name = pmnj-it ]
4037 <ppl> [ name = ppl-it ]
4038 <ptm> [ name = ptm-it ]
4039 <ugm> [ name = ugm-it ]
4040 <m-t|bch|pad|pmn|ugm> { encoding = OT1,
4041 <ppl|ptm> { encoding = {OT1,OT4},
4042 <bch> family = bch,
4043 <pad> family = {pad,padx,padj},
4044 <pmn> family = pmnj,

```

```

4045 <ppi> family = {ppi,pplx,pplj},
4046 <ptm> family = {ptm,ptmx,ptmj},
4047 <ugm> family = ugm,
4048 <! (cmr|ugm)> shape = {it,s1} }
4049 <ugm> shape = it }
4050 <cmr> { }
4051 {
4052 <cmr|ptm> A = {100,50},
4053 <pad|pmn> A = {50, },
4054 <ugm> A = { ,150},
4055 <ppi> A = {50,50},
4056 <ptm> \AE = {100, },
4057 <pad|ppi> \AE = {50, },
4058 <pmn> \AE = { , -50},
4059 <cmr|pad|ppi|ptm> B = {50, },
4060 <pmn> B = {20,-50},
4061 <bch|ppi|ptm|ugm> C = {50, },
4062 <cmr|pad> C = {100, },
4063 <pmn> C = {50,-50},
4064 <cmr|pad|ppi|ptm> D = {50,50},
4065 <pmn> D = {20, },
4066 <cmr|pad|ppi|ptm> E = {50, },
4067 <pmn> E = {20,-50},
4068 <cmr|pad|ptm> F = {100, },
4069 <pmn> F = {10, },
4070 <ppi> F = {50, },
4071 <bch|ppi|ptm|ugm> G = {50, },
4072 <cmr|pad> G = {100, },
4073 <pmn> G = {50,-50},
4074 <cmr|pad|ppi|ptm> H = {50, },
4075 <cmr|pad|ptm> I = {50, },
4076 <pmn> I = {20,-50},
4077 <cmr|ptm> J = {100, },
4078 <pad> J = {50, },
4079 <pmn> J = {20, },
4080 <cmr|pad|ppi|ptm> K = {50, },
4081 <pmn> K = {20, },
4082 <cmr|pad|ppi|ptm> L = {50, },
4083 <pmn> L = {20,50},
4084 <ugm> L = { ,100},
4085 <cmr|ptm> M = {50, },
4086 <pmn> M = { , -30},
4087 <cmr|ptm> N = {50, },
4088 <pmn> N = { , -30},
4089 <bch|pmn|ppi|ptm> O = {50, },
4090 <cmr|pad> O = {100, },
4091 <ugm> O = {70,50},
4092 <pmn|ppi|ptm> \OE = {50, },
4093 <pad> \OE = {100, },
4094 <cmr|pad|ppi|ptm> P = {50, },
4095 <pmn> P = {20,-50},
4096 <bch|pmn|ppi|ptm> Q = {50, },
4097 <cmr|pad> Q = {100, },
4098 <ugm> Q = {70,50},
4099 <cmr|pad|ppi|ptm> R = {50, },
4100 <pmn> R = {20, },
4101 <bch|cmr|pad|ppi|ptm> S = {50, },
4102 <pmn> S = {20,-30},
4103 <bch|cmr|pad|ppi|ptm> $ = {50, },
4104 <pmn> $ = {20,-30},
4105 <bch|pmn|ugm> T = {70, },
4106 <cmr|pad|ppi|ptm> T = {100, },
4107 <cmr|pad|ppi|ptm> U = {50, },

```

```

4108 <pmn> U = {50,-50},
4109 <cmr|pad|pmn|ugm> V = {100, },
4110 <opl|ptm> V = {100,50},
4111 <cmr|pad|pmn|ugm> W = {100, },
4112 <opl> W = {50, },
4113 <ptm> W = {100,50},
4114 <cmr|opl|ptm> X = {50, },
4115 <cmr|ptm> Y = {100, },
4116 <pmn> Y = {50, },
4117 <opl> Y = {100,50},
4118 <pmn> Z = { , -50},
4119 <pmn> d = { , -50},
4120 <pad|pmn> f = { , -100},
4121 <pmn> i = { , -30},
4122 <pmn> j = { , -30},
4123 <pmn> l = { , -100},
4124 <bch> o = {50,50},
4125 <bch> p = { , 50},
4126 <pmn> p = {-50, },
4127 <bch> q = {50, },
4128 <pmn> r = { , 50},
4129 <bch> t = { , 50},
4130 <pmn|ugm> v = {50, },
4131 <bch> w = { , 50},
4132 <pmn|ugm> w = {50, },
4133 <bch> y = { , 50},
4134 <cmr> 0 = {100, },
4135 <bch|ptm> 1 = {150,100},
4136 <cmr> 1 = {200,50},
4137 <pad> 1 = {150, },
4138 <pmn> 1 = {50, },
4139 <opl> 1 = {100, },
4140 <ugm> 1 = {150,150},
4141 <cmr> 2 = {100,-100},
4142 <pad|opl|ptm> 2 = {50, },
4143 <pmn> 2 = {-50, },
4144 <bch> 3 = {50, },
4145 <cmr> 3 = {100,-100},
4146 <pmn> 3 = {-100, },
4147 <ptm> 3 = {100,50},
4148 <bch> 4 = {100, },
4149 <cmr|pad> 4 = {150, },
4150 <opl|ptm> 4 = {50, },
4151 <cmr> 5 = {100, },
4152 <ptm> 5 = {50, },
4153 <bch> 6 = {50, },
4154 <cmr> 6 = {100, },
4155 <bch|pad|ptm> 7 = {100, },
4156 <cmr> 7 = {200,-150},
4157 <pmn> 7 = {20, },
4158 <opl> 7 = {50, },
4159 <cmr> 8 = {50,-50},
4160 <cmr> 9 = {100,-100},
4161 <m-t|cmr|pad|pmn|opl> . = { ,500},
4162 <bch|ptm|ugm> . = { ,700},
4163 <m-t|cmr|pad|pmn|opl> {,}= { ,500},
4164 <bch|ugm> {,}= { ,600},
4165 <ptm> {,}= { ,700},
4166 <m-t|cmr|pad|opl> : = { ,300},
4167 <bch|ugm> : = { ,400},
4168 <pmn> : = { ,200},
4169 <ptm> : = { ,500},
4170 <m-t|cmr|pad|opl> ; = { ,300},

```

```

4171 <bch|ugm> ; = { ,400},
4172 <pmn> ; = { ,200},
4173 <ptm> ; = { ,500},
4174 <ptm> ! = { ,100},
4175 <bch> ? = { ,200},
4176 <ptm> ? = { ,100},
4177 <ttl> ? = { ,300},
4178 <pmn> " = {400,200},
4179 <m-t|pad|pmn|ttl|ptm> & = {50,50},
4180 <bch> & = { ,80},
4181 <cmr> & = {100,50},
4182 <ugm> & = {50,100},
4183 <m-t|cmr|pad|pmn> \% = {100, },
4184 <bch> \% = {50,50},
4185 <ttl|ptm> \% = {100,100},
4186 <ugm> \% = {100,50},
4187 <m-t|pmn|ttl> * = {200,200},
4188 <bch> * = {300,200},
4189 <cmr> * = {400,100},
4190 <pad> * = {500,100},
4191 <ptm|ugm> * = {400,200},
4192 <m-t|cmr|pmn|ttl> + = {150,200},
4193 <bch|ugm> + = {250,250},
4194 <pad|ptm> + = {250,200},
4195 <m-t|pad|pmn|ttl> @ = {50,50},
4196 <bch> @ = {80,50},
4197 <cmr> @ = {200,50},
4198 <ptm> @ = {150,150},
4199 <m-t|bch|ugm> ~ = {150,150},
4200 <cmr|pad|pmn|ttl|ptm> ~ = {200,150},
4201 <ugm> {=} = {200,200},
4202 ( = {200, }, ) = { ,200},
4203 <m-t|cmr|pad|ttl|ptm|ugm> / = {100,200},
4204 <bch> / = { ,150},
4205 <pmn> / = {100,150},
4206 <m-t> - = {300,300},
4207 <bch|pad> - = {300,400},
4208 <pmn> - = {200,300},
4209 <cmr> - = {500,300},
4210 <ttl> - = {300,500},
4211 <ptm> - = {500,500},
4212 <ugm> - = {400,700},
4213 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
4214 <bch> \textendash = {200,300}, \textendash = {150,200},
4215 <cmr> \textendash = {500,300}, \textendash = {400,200},
4216 <pad|ttl|ptm|ugm> \textendash = {300,300}, \textendash = {200,200},
4217 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
4218 <cmr|pad> \textquoteleft = {800,200}, \textquoteright = {800,200},
4219 <ttl> \textquoteleft = {700,400}, \textquoteright = {700,400},
4220 <ptm> \textquoteleft = {800,500}, \textquoteright = {800,500},
4221 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200},
4222 <cmr> \textquotedblleft = {700,100}, \textquotedblright = {500,300},
4223 <pad> \textquotedblleft = {700,200}, \textquotedblright = {700,200},
4224 <ttl> \textquotedblleft = {500,300}, \textquotedblright = {500,300},
4225 <ptm> \textquotedblleft = {700,400}, \textquotedblright = {700,400},
4226 <ugm> \textquotedblleft = {600,200}, \textquotedblright = {600,200},
4227 }
4228
4229 <*cmr>
4230 \SetProtrusion
4231 [ name = cmr-it-OT1,
4232 load = cmr-it ]
4233 { encoding = {OT1,OT4},

```

```

4234     family = cmr,
4235     shape   = it   }
4236 {
4237     \AE = {100, },
4238     \OE = {100, },
4239     "00 = {200,150}, % \Gamma
4240     "01 = {150,100}, % \Delta
4241     "02 = {150, 50}, % \Theta
4242     "03 = {150, 50}, % \Lambda
4243     "04 = {100,100}, % \Xi
4244     "05 = {100,100}, % \Pi
4245     "06 = {100, 50}, % \Sigma
4246     "07 = {200,150}, % \Upsilon
4247     "08 = {150, 50}, % \Phi
4248     "09 = {150,100}, % \Psi
4249     "0A = { 50, 50}, % \Omega
4250 }
4251
4252 </cmr>
4253 \SetProtrusion
4254 <m-t> [ name = T1-it-default,
4255 <bch> [ name = bch-it-T1,
4256 <cmr> [ name = cmr-it-T1,
4257 <pad> [ name = pad-it-T1,
4258 <pmn> [ name = pmnj-it-T1,
4259 <ppl> [ name = ppl-it-T1,
4260 <ptm> [ name = ptm-it-T1,
4261 <ugm> [ name = ugm-it-T1,
4262 <m-t> load = OT1-it ]
4263 <bch> load = bch-it ]
4264 <cmr> load = cmr-it ]
4265 <pmn> load = pmnj-it ]
4266 <pad> load = pad-it ]
4267 <ppl> load = ppl-it ]
4268 <ptm> load = ptm-it ]
4269 <ugm> load = ugm-it ]
4270 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4271 <ptm|ugm> { encoding = {T1},
4272 <bch> family = bch,
4273 <cmr> family = cmr,
4274 <pmn> family = pmnj,
4275 <pad> family = {pad,padx,padj},
4276 <ppl> family = {ppl,pplx,pplj},
4277 <ptm> family = {ptm,ptmx,ptmj},
4278 <ugm> family = ugm,
4279 <!(cmr|ugm)> shape = {it,s1} }
4280 <cmr|ugm> shape = it   }
4281 {
4282 <m-t|bch|pmn> _ = { ,100},
4283 <cmr|ugm> _ = {100,200},
4284 <pad|ppl|ptm> _ = {100,100},
4285 <cmr> \AE = {100, },
4286 <bch> \OE = { 50, },
4287 <cmr> \OE = {100, },
4288 <pmn> 031 = { , -100}, % ffl
4289 <cmr|ptm> 156 = {100, }, % IJ
4290 <pad> 156 = {50, }, % IJ
4291 <pmn> 156 = {20, }, % IJ
4292 <pmn> 188 = { , -30}, % ij
4293 <pmn> \v t = { ,100},
4294 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
4295 <cmr|ugm> \textbackslash = {300,300},
4296 <bch> \textbackslash = {150,150},

```



```

4297 <pmn> \textbackslash = {100,150},
4298 <ugm> \textbar = {200,200},
4299 <cmr> \textquotedblleft = {500,300},
4300 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
4301 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4302 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4303 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
4304 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
4305 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4306 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4307 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4308 <pad> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
4309 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4310 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
4311 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
4312 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4313 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
4314 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
4315 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4316 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
4317 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4318 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
4319 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
4320 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
4321 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
4322 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
4323 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100},
4324 <pmn> \textvisiblespace = {100,100},
4325 }
4326
4327 <*m-t|ptm>
4328 \SetProtrusion
4329 <m-t> [ name = QX-it-default,
4330 <ptm> [ name = ptm-it-QX,
4331 <m-t> load = OT1-it ]
4332 <ptm> load = ptm-it ]
4333 { encoding = {QX},
4334 <ptm> family = {ptm,ptmx,ptmj},
4335 shape = {it,sl} }
4336 {
4337 <ptm> 009 = { , 50}, % fk
4338 {=} = {100,100},
4339 <m-t> \textunderscore = {100,100},
4340 <ptm> \textunderscore = {100,150},
4341 \textbackslash = {100,200},
4342 \quotedblbase = {300,400},
4343 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
4344 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
4345 \textexclamdown = {200, }, \textquestiondown = {200, },
4346 \textbraceleft = {200,100}, \textbraceright = {200,200},
4347 \textless = {100,100}, \textgreater = {100,100},
4348 \textminus = {200,200}, \textdegree = {300,150},
4349 <m-t> \copyright = {100,100}, \textregistered = {100,100},
4350 <ptm> \textregistered = {100,150}, \copyright = {100,150},
4351 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
4352 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
4353 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
4354 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
4355 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
4356 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
4357 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
4358 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
4359 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},

```

```

4360 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
4361 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
4362 <ptm> \textperthousand = { ,50},
4363 }
4364
4365 </m-t|ptm>
4366 <*cmr|bch>
4367 \SetProtrusion
4368 <cmr> [ name = cmr-it-T5,
4369 <cmr> load = cmr-it ]
4370 <bch> [ name = bch-it-T5,
4371 <bch> load = bch-it ]
4372 { encoding = T5,
4373 <bch> family = bch,
4374 <cmr> family = cmr,
4375 shape = it }
4376 {
4377 <bch> - = { ,100},
4378 <cmr> - = {100,200},
4379 <bch> \textbackslash = {150,150},
4380 <cmr> \textbackslash = {300,300},
4381 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4382 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4383 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4384 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4385 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
4386 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4387 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
4388 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
4389 <bch> \textless = {100, }, \textgreater = { ,100},
4390 <cmr> \textless = {300,100}, \textgreater = {200,100},
4391 }
4392
4393 </cmr|bch>

```

Slanted is very similar to italic.

```

4394 <*cmr>
4395 \SetProtrusion
4396 [ name = cmr-sl,
4397 load = cmr-it-OT1 ]
4398 { encoding = {OT1,OT4},
4399 family = cmr,
4400 shape = sl }
4401 {
4402 L = { ,50},
4403 f = { ,-50},
4404 - = {300, },
4405 \textendash = {400, }, \textemdash = {300, },
4406 }
4407
4408 \SetProtrusion
4409 [ name = cmr-sl-T1,
4410 load = cmr-it-T1 ]
4411 { encoding = {T1,LY1},
4412 family = cmr,
4413 shape = sl }
4414 {
4415 L = { ,50},
4416 f = { ,-50},
4417 - = {300, },
4418 \textendash = {400, }, \textemdash = {300, },
4419 }
4420

```

```

4421 \SetProtrusion
4422 [ name = cmr-sl-T5,
4423   load = cmr-it-T5 ]
4424 { encoding = T5,
4425   family = cmr,
4426   shape = sl }
4427 {
4428   L = { ,50},
4429   f = { ,-50},
4430   - = {300, },
4431   \textendash = {400, }, \textendash = {300, },
4432 }
4433
4434 \SetProtrusion
4435 [ name = lmr-it-T1,
4436   load = cmr-it-T1 ]
4437 { encoding = {T1,LV1},
4438   family = lmr,
4439   shape = {it,sl} }
4440 {
4441   \textquotedblleft = {700,100},
4442   \quotedblbase = {600,300},
4443 }
4444

```

Oldstyle numerals are slightly different.

```

4445 \SetProtrusion
4446 [ name = cmr(oldstyle)-it,
4447   load = cmr-it-T1 ]
4448 { encoding = T1,
4449   family = {hfor,cmor},
4450   shape = {it,sl} }
4451 {
4452   1 = {250, 50},
4453   2 = {150,-100},
4454   3 = {100,-50},
4455   4 = {150,150},
4456   6 = {200, },
4457   7 = {200, 50},
4458   8 = {150,-50},
4459   9 = {100, 50},
4460 }
4461
4462 </cmr>
4463 < *pmn >
4464 \SetProtrusion
4465 [ name = pmnx-it,
4466   load = pmnj-it ]
4467 { encoding = OT1,
4468   family = pmnx,
4469   shape = {it,sl} }
4470 {
4471   1 = {100,150},
4472 }
4473
4474 \SetProtrusion
4475 [ name = pmnx-it-T1,
4476   load = pmnj-it-T1 ]
4477 { encoding = {T1,LV1},
4478   family = pmnx,
4479   shape = {it,sl} }
4480 {
4481   1 = {100,150},

```

```

4482 }
4483
4484 </pmm>
4485 <*ptm>
4486 \SetProtrusion
4487 [ name = ptm-it-LY1,
4488   load = ptm-it-T1 ]
4489 { encoding = {LY1},
4490   family = {ptm,ptmx,ptmj},
4491   shape = {it,sl} }
4492 {
4493   - = {100,100},
4494   \texttrademark = {100,100},
4495   \textregistered = {100,100},
4496   \textcopyright = {100,100},
4497   \textdegree = {300,100},
4498   \textminus = {200,200},
4499   \textellipsis = {100,200},
4500   \texteuro = { , },
4501   \textcent = {100,100},
4502   \textquotesingle = {500, },
4503   \textflorin = {100, 70},
4504   \textdagger = {150,150},
4505   \textdaggerdbl = {100,100},
4506   \textbullet = {150,150},
4507   \textonesuperior = {150,100},
4508   \texttwosuperior = {150, 50},
4509   \textthreesuperior = {150, 50},
4510   \textparagraph = {100, },
4511   \textperiodcentered = {500,300},
4512   \textonequarter = { 50, },
4513   \textonehalf = { 50, },
4514   \textplusminus = {100,100},
4515   \textmultiply = {150,150},
4516   \textdivide = {150,150},
4517 }
4518
4519 </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).

```

4520 <!*ugm>
4521 \SetProtrusion
4522 <m-t> [ name = OT1-sc,
4523 <bch> [ name = bch-sc,
4524 <cmr> [ name = cmr-sc-OT1,
4525 <pad> [ name = pad-sc,
4526 <pmn> [ name = pmnj-sc,
4527 <ppl> [ name = ppl-sc,
4528 <ptm> [ name = ptm-sc,
4529 <m-t> load = default ]
4530 <bch> load = bch-default ]
4531 <cmr> load = cmr-OT1 ]
4532 <pad> load = pad-default ]
4533 <pmn> load = pmnj-default ]
4534 <ppl> load = ppl-default ]
4535 <ptm> load = ptm-default ]
4536 <m-t|bch|pad|pmn> { encoding = OT1,

```

```

4537 <cmr|ppl|ptm> { encoding = {OT1,OT4},
4538 <bch> family = bch,
4539 <cmr> family = cmr,
4540 <pad> family = {pad,padx,padj},
4541 <pmn> family = pmnj,
4542 <ppl> family = {ppl,pplx,pplj},
4543 <ptm> family = {ptm,ptmx,ptmj},
4544 shape = sc }
4545 {
4546 a = {50,50},
4547 <cmr|pad|ppl|ptm> \ae = {50, },
4548 <bch|pmn> c = {50, },
4549 <bch|pad|pmn> d = { ,50},
4550 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
4551 <bch|pad|pmn> g = {50, },
4552 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
4553 <bch> j = {100, },
4554 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
4555 <ptm> l = { ,80},
4556 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
4557 <ptm> 013 = { ,80}, % fl
4558 <bch|pad|pmn> o = {50,50},
4559 <pad|pmn> \oe = {50, },
4560 <ppl> p = { 0, 0},
4561 <bch|pad|pmn> q = {50,70},
4562 <ppl> q = { 0, },
4563 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
4564 t = {50,50},
4565 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50},
4566 <ptm> y = {80,80},
4567 }
4568
4569 \SetProtrusion
4570 <m-t> [ name = T1-sc,
4571 <bch> [ name = bch-sc-T1,
4572 <cmr> [ name = cmr-sc-T1,
4573 <pad> [ name = pad-sc-T1,
4574 <pmn> [ name = pmnj-sc-T1,
4575 <ppl> [ name = ppl-sc-T1,
4576 <ptm> [ name = ptm-sc-T1,
4577 <m-t> load = T1-default ]
4578 <bch> load = bch-T1 ]
4579 <cmr> load = cmr-T1 ]
4580 <pad> load = pad-T1 ]
4581 <pmn> load = pmnj-T1 ]
4582 <ppl> load = ppl-T1 ]
4583 <ptm> load = ptm-T1 ]
4584 { encoding = {T1,LY1},
4585 <bch> family = bch,
4586 <cmr> family = cmr,
4587 <pad> family = {pad,padx,padj},
4588 <pmn> family = pmnj,
4589 <ppl> family = {ppl,pplx,pplj},
4590 <ptm> family = {ptm,ptmx,ptmj},
4591 shape = sc }
4592 {
4593 a = {50,50},
4594 <cmr|pad|ppl|ptm> \ae = {50, },
4595 <bch|pmn> c = {50, },
4596 <bch|pad|pmn> d = { ,50},
4597 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
4598 <bch|pad|pmn> g = {50, },
4599 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },

```

```

4600 <bch> j = {100, },
4601 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
4602 <ptm> l = { ,80},
4603 <m-t|bch|cmr|pad|pmn|ppl> 029 = { ,50}, % f1
4604 <ptm> 029 = { ,80}, % f1
4605 <bch|pad|pmn> o = {50,50},
4606 <bch|pad|pmn> \oe = {50, },
4607 <ppl> p = { 0, 0},
4608 <bch|pad|pmn> q = {50,70},
4609 <ppl> q = { 0, },
4610 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
4611 t = {50,50},
4612 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50},
4613 <ptm> y = {80,80},
4614 }
4615
4616 </!ugm>
4617 <*-t>
4618 \SetProtrusion
4619 [ name = QX-sc,
4620 load = QX-default ]
4621 { encoding = QX,
4622 shape = sc }
4623 {
4624 a = {50,50},
4625 f = { ,50},
4626 j = {50, },
4627 l = { ,50},
4628 013 = { ,50}, % f1
4629 r = { , 0},
4630 t = {50,50},
4631 y = {50,50},
4632 }
4633
4634 </m-t>
4635 <*-cmr|bch>
4636 \SetProtrusion
4637 <bch> [ name = bch-sc-T5,
4638 <bch> load = bch-T5 ]
4639 <cmr> [ name = cmr-sc-T5,
4640 <cmr> load = cmr-T5 ]
4641 { encoding = T5,
4642 <bch> family = bch,
4643 <cmr> family = cmr,
4644 shape = sc }
4645 {
4646 a = {50,50},
4647 <bch> c = {50, },
4648 <bch> d = { ,50},
4649 f = { ,50},
4650 <bch> g = {50, },
4651 <bch> j = {100, },
4652 <cmr> j = {50, },
4653 l = { ,50},
4654 <bch> o = {50,50},
4655 <bch> q = { 0, },
4656 <cmr> r = { , 0},
4657 t = {50,50},
4658 y = {50,50},
4659 }
4660
4661 </cmr|bch>
4662 <*-pmn>

```

```

4663 \SetProtrusion
4664 [ name = pmnx-sc,
4665   load = pmnj-sc ]
4666 { encoding = OT1,
4667   family = pmnx,
4668   shape = sc }
4669 {
4670   1 = {230,180},
4671 }
4672
4673 \SetProtrusion
4674 [ name = pmnx-sc-T1,
4675   load = pmnj-sc-T1 ]
4676 { encoding = {T1,LY1},
4677   family = pmnx,
4678   shape = sc }
4679 {
4680   1 = {230,180},
4681 }
4682

```

#### 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`.

```

4683 \SetProtrusion
4684 [ name = pmnj-scit,
4685   load = pmnj-it ]
4686 { encoding = OT1,
4687   family = pmnj,
4688   shape = {scit,si} }
4689 {
4690   a = {50, },
4691   \ae = { , -50},
4692   b = {20, -50},
4693   c = {50, -50},
4694   d = {20, 0},
4695   e = {20, -50},
4696   f = {10, 0},
4697   012 = {10, -50}, % fi
4698   013 = {10, -50}, % fl
4699   014 = {10, -50}, % ffi
4700   015 = {10, -50}, % ffl
4701   g = {50, -50},
4702   i = {20, -50},
4703   j = {20, 0},
4704   k = {20, },
4705   l = {20, 50},
4706   m = { , -30},
4707   n = { , -30},
4708   o = {50, },
4709   \oe = {50, -50},
4710   p = {20, -50},
4711   q = {50, },
4712   r = {20, 0},
4713   s = {20, -30},
4714   t = {70, },
4715   u = {50, -50},
4716   v = {100, },
4717   w = {100, },
4718   y = {50, },

```

```
4719     z = { , -50},
4720 }
4721
4722 \SetProtrusion
4723 [ name      = pmnj-scit-T1,
4724   load      = pmnj-it-T1 ]
4725 { encoding = {T1,LV1},
4726   family   = pmnj,
4727   shape    = {scit,si} }
4728 {
4729   a = {50, },
4730 \ae = { , -50},
4731   b = {20,-50},
4732   c = {50,-50},
4733   d = {20, 0},
4734   e = {20,-50},
4735   f = {10, 0},
4736   028 = {10,-50}, % fi
4737   029 = {10,-50}, % fl
4738   030 = {10,-50}, % ffi
4739   031 = {10,-50}, % ff1
4740   g = {50,-50},
4741   i = {20,-50},
4742   188 = {20, 0}, % ij
4743   j = {20, 0},
4744   k = {20, },
4745   l = {20,50},
4746   m = { , -30},
4747   n = { , -30},
4748   o = {50, },
4749 \oe = {50,-50},
4750   p = {20,-50},
4751   q = {50, },
4752   r = {20, 0},
4753   s = {20,-30},
4754   t = {70, },
4755   u = {50,-50},
4756   v = {100, },
4757   w = {100, },
4758   y = {50, },
4759   z = { , -50},
4760 }
4761
4762 \SetProtrusion
4763 [ name      = pmnx-scit,
4764   load      = pmnj-scit ]
4765 { encoding = OT1,
4766   family   = pmnx,
4767   shape    = {scit,si} }
4768 {
4769   1 = {100,150},
4770 }
4771
4772 \SetProtrusion
4773 [ name      = pmnx-scit-T1,
4774   load      = pmnj-scit-T1 ]
4775 { encoding = {T1,LV1},
4776   family   = pmnx,
4777   shape    = {scit,si} }
4778 {
4779   1 = {100,150},
4780 }
4781
```



4782  $\langle /pmn \rangle$

### 14.7.5 textcomp

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

4783 \SetProtrusion
4784  $\langle m-t \rangle$  [ name = textcomp ]
4785  $\langle bch \rangle$  [ name = bch-textcomp ]
4786  $\langle cmr \rangle$  [ name = cmr-textcomp ]
4787  $\langle pad \rangle$  [ name = pad-textcomp ]
4788  $\langle pmn \rangle$  [ name = pmn-textcomp ]
4789  $\langle ppl \rangle$  [ name = ppl-textcomp ]
4790  $\langle ptm \rangle$  [ name = ptm-textcomp ]
4791  $\langle ugm \rangle$  [ name = ugm-textcomp ]
4792  $\langle m-t \rangle$  { encoding = TS1 }
4793  $\langle /m-t \rangle$  { encoding = TS1,
4794  $\langle bch \rangle$  family = bch }
4795  $\langle cmr \rangle$  family = cmr }
4796  $\langle pad \rangle$  family = { pad,padx,padj } }
4797  $\langle pmn \rangle$  family = { pmnx,pmnj } }
4798  $\langle ppl \rangle$  family = { ppl,pplx,pplj } }
4799  $\langle ptm \rangle$  family = { ptm,ptmx,ptmj } }
4800  $\langle ugm \rangle$  family = ugm }
4801 {
4802  $\langle cmr \rangle$  \textquotestraightbase = {300,300},
4803  $\langle pad|pmn \rangle$  \textquotestraightbase = {400,400},
4804  $\langle cmr|pmn \rangle$  \textquotestraightdblbase = {300,300},
4805  $\langle pad \rangle$  \textquotestraightdblbase = {400,400},
4806  $\langle bch|cmr|pad|pmn|ugm \rangle$  \texttwelveudash = {200,200},
4807  $\langle bch|cmr|pad|pmn \rangle$  \textthreequartersemdash = {150,150},
4808  $\langle ugm \rangle$  \textthreequartersemdash = {200,200},
4809  $\langle cmr|pmn \rangle$  \textquotesingle = {300,400},
4810  $\langle pad \rangle$  \textquotesingle = {400,500},
4811  $\langle ptm \rangle$  \textquotesingle = {500,500},
4812  $\langle ugm \rangle$  \textquotesingle = {300,500},
4813  $\langle bch|cmr|pmn \rangle$  \textasteriskcentered = {200,300},
4814  $\langle pad \rangle$  \textasteriskcentered = {300,300},
4815  $\langle ugm \rangle$  \textasteriskcentered = {100,200},
4816  $\langle pmn \rangle$  \textfractionsolidus = {-200,-200},
4817  $\langle cmr \rangle$  \textoneoldstyle = {100,100},
4818  $\langle pmn \rangle$  \textoneoldstyle = { , 50},
4819  $\langle cmr \rangle$  \textthreeoldstyle = { , 50},
4820  $\langle pad|pmn \rangle$  \textthreeoldstyle = { 50, },
4821  $\langle cmr \rangle$  \textfouroldstyle = { 50, 50},
4822  $\langle pad|pmn \rangle$  \textfouroldstyle = { 50, },
4823  $\langle cmr|pad|pmn \rangle$  \textsevenoldstyle = { 50, 80},
4824  $\langle cmr \rangle$  \textlangle = {400, },
4825  $\langle cmr \rangle$  \extrangle = { ,400},
4826  $\langle m-t|bch|pmn|ptm \rangle$  \textminus = {200,200},
4827  $\langle cmr|pad|ppl \rangle$  \textminus = {300,300},
4828  $\langle ugm \rangle$  \textminus = {250,300},
4829  $\langle bch|pad|pmn \rangle$  \textlbrackdbl = {100, },
4830  $\langle bch|pad|pmn \rangle$  \extrbrackdbl = { ,100},
4831  $\langle pmn \rangle$  \textasciigrave = {200,500},
4832  $\langle bch|cmr|pad|pmn \rangle$  \texttildelow = {200,250},
4833  $\langle pmn \rangle$  \textasciibreve = {300,400},
4834  $\langle pmn \rangle$  \textasciicaron = {300,400},
4835  $\langle pmn \rangle$  \textacutedbl = {200,300},
4836  $\langle pmn \rangle$  \textgravedbl = {150,300},
4837  $\langle bch|pmn|ugm \rangle$  \textdagger = { 80, 80},

```

```

4838 <cmr|pad> \textdagger = {100,100},
4839 <ptm> \textdagger = {150,150},
4840 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
4841 <ptm> \textdaggerdbl = {100,100},
4842 <bch> \textbardbl = {100,100},
4843 <ugm> \textbardbl = {150,150},
4844 <bch> \textbullet = {200,200},
4845 <cmr|pad|pmn> \textbullet = { ,100},
4846 <ptm> \textbullet = {150,150},
4847 <ugm> \textbullet = { 50,100},
4848 <bch|cmr|pmn> \textcelsius = { 50, },
4849 <pad> \textcelsius = { 80, },
4850 <bch> \textflorin = { 50, 50},
4851 <pad|ugm> \textflorin = { ,100},
4852 <pmn> \textflorin = { 50,100},
4853 <ptm> \textflorin = { 50, 70},
4854 <cmr> \textcolonmonetary = { , 50},
4855 <pad|pmn> \textcolonmonetary = { 50, },
4856 <pmn> \textinterrobang = { ,100},
4857 <pmn> \textinterrobangdown = {100, },
4858 <m-t|pad|ptm> \texttrademark = {100,100},
4859 <bch> \texttrademark = {150,150},
4860 <cmr|ppl> \texttrademark = {200,200},
4861 <pmn> \texttrademark = { 50, 50},
4862 <ugm> \texttrademark = {100,150},
4863 <bch|ugm> \textcent = { 50, },
4864 <ptm> \textcent = {100,100},
4865 <bch> \textsterling = { 50, },
4866 <ugm> \textsterling = { , 50},
4867 <bch> \textbrokenbar = {200,200},
4868 <ugm> \textbrokenbar = {200,300},
4869 <pmn> \textasciidieresis = {300,400},
4870 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
4871 <pmn> \textcopyright = {100,150},
4872 <ppl> \textcopyright = {200,200},
4873 <bch|cmr|ugm> \textordfeminine = {100,200},
4874 <pad|pmn> \textordfeminine = {200,200},
4875 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
4876 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
4877 <pmn> \textregistered = { 50,150},
4878 <ppl> \textregistered = {200,200},
4879 <pmn> \textasciimacron = {150,200},
4880 <m-t|ppl|ptm> \textdegree = {300,300},
4881 <bch> \textdegree = {150,200},
4882 <cmr|pad> \textdegree = {400,400},
4883 <pmn> \textdegree = {150,400},
4884 <ugm> \textdegree = {200,200},
4885 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
4886 <ptm> \textpm = { 50, 80},
4887 <bch|ugm> \texttwosuperior = {100,200},
4888 <cmr> \texttwosuperior = { 50,100},
4889 <pad|pmn> \texttwosuperior = {200,200},
4890 <ptm> \texttwosuperior = { 50, 50},
4891 <bch|ugm> \textthreesuperior = {100,200},
4892 <cmr> \textthreesuperior = { 50,100},
4893 <pad|pmn> \textthreesuperior = {200,200},
4894 <ptm> \textthreesuperior = { 50, 50},
4895 <pmn> \textasciiaacute = {300,400},
4896 <bch|ugm> \textmu = { ,100},
4897 <bch|pad|pmn> \textparagraph = { ,100},
4898 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
4899 <ptm> \textperiodcentered = {300,300},
4900 <ugm> \textperiodcentered = {200,500},

```

```

4901 <bch|ugm> \textonesuperior = {200,300},
4902 <cmr|pad|pmn> \textonesuperior = {200,200},
4903 <ptm> \textonesuperior = {100,100},
4904 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
4905 <cmr> \textordmasculine = {100,200},
4906 <bch|cmr|pmn> \texteuro = {100, },
4907 <pad> \texteuro = { 50,100},
4908 <bch|ptm> \textttimes = {100,100},
4909 <cmr> \textttimes = {150,250},
4910 <pad> \textttimes = {100,150},
4911 <pmn> \textttimes = { 70,100},
4912 <ugm> \textttimes = {200,300},
4913 <bch|pad|pmn> \textdiv = {150,200},
4914 <cmr> \textdiv = {150,250},
4915 <ptm> \textdiv = { 50,100},
4916 <ugm> \textdiv = {200,300},
4917 <ptm> \textperthousand = { ,50},
4918 <ugm> \textsection = { ,100},
4919 <ugm> \textonehalf = { 50,100},
4920 <ugm> \textonequarter = { 50,100},
4921 <ugm> \textthreequarters = { 50,100},
4922 <ugm> \textsurd = { ,100},

```

Remaining slots in the source file.

```

4923 }
4924
4925 (*cmr|pad|pmn|ugm)
4926 \SetProtrusion
4927 <cmr> [ name = cmr-textcomp-it ]
4928 <pad> [ name = pad-textcomp-it ]
4929 <pmn> [ name = pmn-textcomp-it ]
4930 <ugm> [ name = ugm-textcomp-it ]
4931 { encoding = TS1,
4932 <cmr> family = cmr,
4933 <pad> family = {pad,padx,padj},
4934 <pmn> family = {pmnx,pmnj},
4935 <ugm> family = ugm,
4936 <!ugm> shape = {it,s1} }
4937 <ugm> shape = it }
4938 {
4939 <cmr> \textquotestraightbase = {300,600},
4940 <pad|pmn> \textquotestraightbase = {400,400},
4941 <cmr> \textquotestraightdblbase = {300,600},
4942 <pad> \textquotestraightdblbase = {300,400},
4943 <pmn> \textquotestraightdblbase = {300,300},
4944 \texttwelveudash = {200,200},
4945 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
4946 <ugm> \textthreequartersemdash = {200,200},
4947 <cmr> \textquotesingle = {600,300},
4948 <pad> \textquotesingle = {800,100},
4949 <pmn> \textquotesingle = {300,200},
4950 <ugm> \textquotesingle = {500,500},
4951 <cmr> \textasteriskcentered = {300,200},
4952 <pad> \textasteriskcentered = {500,100},
4953 <pmn> \textasteriskcentered = {200,300},
4954 <ugm> \textasteriskcentered = {300,150},
4955 <pmn> \textfractionsolidus = {-200,-200},
4956 <cmr> \textoneoldstyle = {100, 50},
4957 <pad> \textoneoldstyle = {100, },
4958 <pmn> \textoneoldstyle = { 50, },
4959 <pad> \texttwooldstyle = { 50, },
4960 <pmn> \texttwooldstyle = {-50, },
4961 <cmr> \textthreeoldstyle = {100, 50},

```

```

4962 <pmn> \textthreeoldstyle = {-100, },
4963 <cmr> \textfouroldstyle = { 50, 50},
4964 <pad> \textfouroldstyle = { 50,100},
4965 <cmr> \textsevenoldstyle = { 50, 80},
4966 <pad> \textsevenoldstyle = { 50, },
4967 <pmn> \textsevenoldstyle = { 20, },
4968 <cmr> \textlangle = {400, },
4969 <cmr> \textrangle = { ,400},
4970 <cmr|pad> \textminus = {300,300},
4971 <pmn> \textminus = {200,200},
4972 <ugm> \textminus = {250,300},
4973 <pad|pmn> \textlbrackdbl = {100, },
4974 <pad|pmn> \textrbrackdbl = { ,100},
4975 <pmn> \textasciigrave = {300,300},
4976 <cmr|pad|pmn> \texttildelow = {200,250},
4977 <pmn> \textasciibreve = {300,300},
4978 <pmn> \textasciicaron = {300,300},
4979 <pmn> \textacutedbl = {200,300},
4980 <pmn> \textgravedbl = {150,300},
4981 <cmr> \textdagger = {100,100},
4982 <pad> \textdagger = {200,100},
4983 <pmn> \textdagger = { 80, 50},
4984 <ugm> \textdagger = { 80, 80},
4985 <cmr|pad> \textdaggerdbl = { 80, 80},
4986 <pmn> \textdaggerdbl = { 80, 50},
4987 <ugm> \textbardbl = {150,150},
4988 <cmr> \textbullet = {200,100},
4989 <pad> \textbullet = {300, },
4990 <pmn> \textbullet = { 30, 70},
4991 <ugm> \textbullet = { 50,100},
4992 <cmr> \textcelsius = {100, },
4993 <pad> \textcelsius = {200, },
4994 <pmn> \textcelsius = { 50,-50},
4995 <pad> \textflorin = {100, },
4996 <pmn> \textflorin = { 50,100},
4997 <ugm> \textflorin = { ,100},
4998 <cmr> \textcolonmonetary = {150, },
4999 <pad> \textcolonmonetary = {100, },
5000 <pmn> \textcolonmonetary = { 50,-50},
5001 <cmr|pad> \texttrademark = {200, },
5002 <pmn> \texttrademark = { 50,100},
5003 <ugm> \texttrademark = {150, 50},
5004 <ugm> \textcent = { 50, },
5005 <ugm> \textsterling = { , 50},
5006 <ugm> \textbrokenbar = {200,300},
5007 <pmn> \textasciidieresis = {300,200},
5008 <cmr> \textcopyright = {100, },
5009 <pad> \textcopyright = {200,100},
5010 <pmn> \textcopyright = {100,150},
5011 <ugm> \textcopyright = {300, },
5012 <cmr> \textordfeminine = {100,100},
5013 <pmn> \textordfeminine = {200,200},
5014 <ugm> \textordfeminine = {100,200},
5015 <cmr|pad> \textlnot = {300, },
5016 <pmn|ugm> \textlnot = {200, },
5017 <cmr> \textregistered = {100, },
5018 <pad> \textregistered = {200,100},
5019 <pmn> \textregistered = { 50,150},
5020 <ugm> \textregistered = {300, },
5021 <pmn> \textasciimacron = {150,200},
5022 <cmr|pad> \textdegree = {500,100},
5023 <pmn> \textdegree = {150,150},
5024 <ugm> \textdegree = {300,200},

```

```

5025 <cmr> \textpm = {150,100},
5026 <pad> \textpm = {200,150},
5027 <pmn|ugm> \textpm = {150,200},
5028 <cmr> \textonesuperior = {400, },
5029 <pad> \textonesuperior = {300,100},
5030 <pmn> \textonesuperior = {200,100},
5031 <ugm> \textonesuperior = {300,300},
5032 <cmr> \texttwosuperior = {400, },
5033 <pad> \texttwosuperior = {300, },
5034 <pmn> \texttwosuperior = {200,100},
5035 <ugm> \texttwosuperior = {300,200},
5036 <cmr> \textthreesuperior = {400, },
5037 <pad> \textthreesuperior = {300, },
5038 <pmn> \textthreesuperior = {200,100},
5039 <ugm> \textthreesuperior = {300,200},
5040 <ugm> \textmu = { ,100},
5041 <pmn> \textasciicute = {300,200},
5042 <cmr> \textparagraph = {200, },
5043 <pmn> \textparagraph = { ,100},
5044 <cmr> \textperiodcentered = {500,500},
5045 <pad|pmn|ugm> \textperiodcentered = {300,400},
5046 <cmr> \textordmasculine = {100,100},
5047 <pmn> \textordmasculine = {200,200},
5048 <ugm> \textordmasculine = {300,200},
5049 <cmr> \texteuro = {200, },
5050 <pad> \texteuro = {100, },
5051 <pmn> \texteuro = {100,-50},
5052 <cmr> \texttimes = {200,200},
5053 <pad> \texttimes = {200,100},
5054 <pmn> \texttimes = { 70,100},
5055 <ugm> \texttimes = {200,300},
5056 <cmr|pad> \textdiv = {200,200},
5057 <pmn> \textdiv = {150,200},
5058 <ugm> \textdiv = {200,300},
5059 <ugm> \textsection = { ,200},
5060 <ugm> \textonehalf = { 50,100},
5061 <ugm> \textonequarter = { 50,100},
5062 <ugm> \textthreequarters = { 50,100},
5063 <ugm> \textsurd = { ,100},
5064 }
5065
5066 </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}

```

```

5067 <*cmr>
5068 \SetProtrusion
5069 [ name = cmr-math-letters ]
5070 { encoding = OML,
5071   family = cmm,
5072   series = {m,b},
5073   shape = it }
5074 {
5075   A = {100, 50}, % \mathnormal
5076   B = { 50, },
5077   C = { 50, },
5078   D = { 50, 50},
5079   E = { 50, },
5080   F = {100, 50},
5081   G = { 50, 50},
5082   H = { 50, 50},
5083   I = { 50, 50},
5084   J = {150, 50},
5085   K = { 50,100},
5086   L = { 50, 50},
5087   M = { 50, },
5088   N = { 50, },
5089   O = { 50, },
5090   P = { 50, },
5091   Q = { 50, 50},
5092   R = { 50, },
5093   S = { 50, },
5094   T = { 50,100},
5095   U = { 50, 50},
5096   V = {100,100},
5097   W = { 50,100},
5098   X = { 50,100},
5099   Y = {100,100},
5100   f = {100,100},
5101   h = { ,100},
5102   i = { , 50},
5103   j = { , 50},
5104   k = { , 50},
5105   r = { , 50},
5106   v = { , 50},
5107   w = { , 50},
5108   x = { , 50},
5109   "OB = { 50,100}, % \alpha
5110   "OC = { 50, 50}, % \beta
5111   "OD = {200,150}, % \gamma
5112   "OE = { 50, 50}, % \delta
5113   "OF = { 50, 50}, % \epsilon
5114   "10 = { 50,150}, % \zeta
5115   "12 = { 50, }, % \theta
5116   "13 = { ,100}, % \iota
5117   "14 = { ,100}, % \kappa
5118   "15 = {100, 50}, % \lambda
5119   "16 = { , 50}, % \mu
5120   "17 = { , 50}, % \nu
5121   "18 = { , 50}, % \xi
5122   "19 = { 50,100}, % \pi
5123   "1A = { 50, 50}, % \rho
5124   "1B = { ,150}, % \sigma
5125   "1C = { 50,150}, % \tau
5126   "1D = { 50, 50}, % \upsilon
5127   "1F = { 50,100}, % \chi
5128   "20 = { 50, 50}, % \psi
5129   "21 = { , 50}, % \omega

```

```

5130 "22 = { , 50}, % \varepsilon
5131 "23 = { , 50}, % \vartheta
5132 "24 = { , 50}, % \varpi
5133 "25 = {100, }, % \varrho
5134 "26 = {100,100}, % \varsigma
5135 "27 = { 50, 50}, % \varphi
5136 "28 = {100,100}, % \leftharpoonup
5137 "29 = {100,100}, % \leftharpoondown
5138 "2A = {100,100}, % \rightharpoonup
5139 "2B = {100,100}, % \rightharpoondown
5140 "2C = {300,200}, % \lhook
5141 "2D = {200,300}, % \rhook
5142 "2E = { ,100}, % \triangleright
5143 "2F = {100, }, % \triangleleft
5144 "3A = { ,500}, % ., \ldotp
5145 "3B = { ,500}, % ,
5146 "3C = {200,100}, % <
5147 "3D = {300,400}, % /
5148 "3E = {100,200}, % >
5149 "3F = {200,200}, % \star
5150 "5B = { ,100}, % \flat
5151 "5E = {200,200}, % \smile
5152 "5F = {200,200}, % \frown
5153 "7C = {100, }, % \jmath
5154 "7D = { ,100}, % \wp

```

Remaining slots in the source file.

```

5155 }
5156

```

Math font 'symbols' (also used for the `\mathcal` alphabet) is declared as:

```

\DeclareSymbolFont{symbols} {OMS}{cmsy}{m}{n}
\SetSymbolFont{symbols} {bold}{OMS}{cmsy}{b}{n}

```

```

5157 \SetProtrusion
5158 [ name = cmr-math-symbols ]
5159 { encoding = OMS,
5160 family = cmsy,
5161 series = {m,b},
5162 shape = n }
5163 {
5164 A = {150, 50}, % \mathcal
5165 C = { ,100},
5166 D = { , 50},
5167 F = { 50,150},
5168 I = { ,100},
5169 J = {100,150},
5170 K = { ,100},
5171 L = {100, },
5172 M = { 50, 50},
5173 N = { 50,100},
5174 P = { , 50},
5175 Q = { 50, },
5176 R = { , 50},
5177 T = { 50,150},
5178 V = { 50, 50},
5179 W = { , 50},
5180 X = {100,100},
5181 Y = {100, },
5182 Z = {100,150},
5183 "00 = {300,300}, % -
5184 "01 = { ,700}, % \cdot, \cdotp

```

```

5185 "02 = {150,250}, % \times
5186 "03 = {150,250}, % *, \ast
5187 "04 = {200,300}, % \div
5188 "05 = {150,250}, % \diamond
5189 "06 = {200,200}, % \pm
5190 "07 = {200,200}, % \mp
5191 "08 = {100,100}, % \oplus
5192 "09 = {100,100}, % \ominus
5193 "0A = {100,100}, % \otimes
5194 "0B = {100,100}, % \oslash
5195 "0C = {100,100}, % \odot
5196 "0D = {100,100}, % \bigcirc
5197 "0E = {100,100}, % \circ
5198 "0F = {100,100}, % \bullet
5199 "10 = {100,100}, % \asymp
5200 "11 = {100,100}, % \equiv
5201 "12 = {200,100}, % \subseteq
5202 "13 = {100,200}, % \supseteq
5203 "14 = {200,100}, % \leq
5204 "15 = {100,200}, % \geq
5205 "16 = {200,100}, % \preceq
5206 "17 = {100,200}, % \succeq
5207 "18 = {200,200}, % \sim
5208 "19 = {150,150}, % \approx
5209 "1A = {200,100}, % \subset
5210 "1B = {100,200}, % \supset
5211 "1C = {200,100}, % \ll
5212 "1D = {100,200}, % \gg
5213 "1E = {300,100}, % \prec
5214 "1F = {100,300}, % \succ
5215 "20 = {100,200}, % \leftarrow
5216 "21 = {200,100}, % \rightarrow
5217 "22 = {100,100}, % \uparrow
5218 "23 = {100,100}, % \downarrow
5219 "24 = {100,100}, % \leftrightarrows
5220 "25 = {100,100}, % \nearrow
5221 "26 = {100,100}, % \searrow
5222 "27 = {100,100}, % \simeq
5223 "28 = {100,100}, % \Leftarrow
5224 "29 = {100,100}, % \Rightarrow
5225 "2A = {100,100}, % \Uparrow
5226 "2B = {100,100}, % \Downarrow
5227 "2C = {100,100}, % \Leftrightarrow
5228 "2D = {100,100}, % \nrightarrow
5229 "2E = {100,100}, % \swarrow
5230 "2F = { ,100}, % \propto
5231 "30 = { ,400}, % \prime
5232 "31 = {100,100}, % \infty
5233 "32 = {150,100}, % \in
5234 "33 = {100,150}, % \ni
5235 "34 = {100,100}, % \triangle, \bigtriangleup
5236 "35 = {100,100}, % \bigtriangledown
5237 "38 = { ,100}, % \forall
5238 "39 = {100, }, % \exists
5239 "3A = {200, }, % \neg
5240 "3E = {200,200}, % \top
5241 "3F = {200,200}, % \bot, \perp
5242 "5E = {100,200}, % \wedge
5243 "5F = {100,200}, % \vee
5244 "60 = { ,300}, % \vdash
5245 "61 = {300, }, % \dashv
5246 "62 = {100,100}, % \lfloor
5247 "63 = {100,100}, % \rfloor

```



```

5248 "64 = {100,100}, % \lceil
5249 "65 = {100,100}, % \rceil
5250 "66 = {150, }, % \lbrace
5251 "67 = { ,150}, % \rbrace
5252 "68 = {400, }, % \langle
5253 "69 = { ,400}, % \rangle
5254 "6C = {100,100}, % \updownarrow
5255 "6D = {100,100}, % \Updownarrow
5256 "6E = {100,300}, % \, \backslash, \setminus
5257 "72 = {100,100}, % \nabla
5258 "79 = {200,200}, % \dagger
5259 "7A = {100,100}, % \ddagger
5260 "7B = {100, }, % \mathparagraph
5261 "7C = {100,100}, % \clubsuit
5262 "7D = {100,100}, % \diamondsuit
5263 "7E = {100,100}, % \heartsuit
5264 "7F = {100,100}, % \spadesuit

```

Remaining slots in the source file.

```

5265 }
5266

```

We don't bother about 'largsymbols', since it will only be used in display math, where protrusion doesn't work anyway. It's declared as:

```
\DeclareSymbolFont{largsymbols}{OMX}{cmex}{m}{n}
```

```

5267 </cmr>
5268 </cfg-t>

```

### 14.7.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
5269 <*cfg-u>
```

Symbol font 'a'.

```

5270 <*msa>
5271 \SetProtrusion
5272 [ name = AMS-a ]
5273 { encoding = U,
5274   family = msa }
5275 {
5276 "05 = {150,250}, % \centerdot
5277 "06 = {100,100}, % \lozenge
5278 "07 = { 50, 50}, % \blacklozenge
5279 "08 = { 50, 50}, % \circlearrowright
5280 "09 = { 50, 50}, % \circlearrowleft
5281 "0A = {100,100}, % \rightleftharpoons
5282 "0B = {100,100}, % \leftrightharpoons
5283 "0D = {-50,200}, % \Vdash
5284 "0E = {-50,200}, % \Vvdash
5285 "0F = {-70,150}, % \vDash
5286 "10 = {100,150}, % \twoheadrightarrow
5287 "11 = {100,150}, % \twoheadleftarrow
5288 "12 = { 50,100}, % \leftleftarrows
5289 "13 = { 50, 80}, % \rightrightarrows
5290 "14 = {120,120}, % \upuparrows
5291 "15 = {120,120}, % \downdownarrows
5292 "16 = {200,200}, % \upharpoonright
5293 "17 = {200,200}, % \downharpoonright
5294 "18 = {200,200}, % \upharpoonleft

```

```

5295 "19 = {200,200}, % \downharpoonleft
5296 "1A = { 80,100}, % \rightarrowtail
5297 "1B = { 80,100}, % \leftarrowtail
5298 "1C = { 50, 50}, % \leftrightarrows
5299 "1D = { 50, 50}, % \rightleftarrows
5300 "1E = {250,  }, % \Lsh
5301 "1F = {  ,250}, % \Rsh
5302 "20 = {100,100}, % \rightsquigarrow
5303 "21 = {100,100}, % \leftrightsquigarrow
5304 "22 = {100, 50}, % \looparrowleft
5305 "23 = { 50,100}, % \looparrowright
5306 "24 = { 50, 80}, % \circeq
5307 "25 = {  ,100}, % \succsim
5308 "26 = {  ,100}, % \gtrsim
5309 "27 = {  ,100}, % \gtrapprox
5310 "28 = {150, 50}, % \multimap
5311 "2B = {100,150}, % \doteqdot
5312 "2C = {100,150}, % \triangleq
5313 "2D = {100, 50}, % \precsim
5314 "2E = {100, 50}, % \lessim
5315 "2F = { 50, 50}, % \lessapprox
5316 "30 = {100, 50}, % \eqslantless
5317 "31 = { 50, 50}, % \eqslantgtr
5318 "32 = {100, 50}, % \curlyeqprec
5319 "33 = { 50,100}, % \curlyeqsucc
5320 "34 = {100, 50}, % \preccurlyeq
5321 "36 = { 50,  }, % \leqslant
5322 "38 = {  , 50}, % \backprime
5323 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
5324 "3C = { 50,100}, % \succcurlyeq
5325 "3E = {  , 50}, % \geqslant
5326 "40 = {  , 50}, % \sqsubset
5327 "41 = { 50,  }, % \sqsupset
5328 "42 = {  ,150}, % \vartriangleright, \rhd
5329 "43 = {150,  }, % \vartriangleleft, \lhd
5330 "44 = {  ,100}, % \trianglerighteq, \unrhd
5331 "45 = {100,  }, % \trianglelefteq, \unlhd
5332 "46 = {100,100}, % \bigstar
5333 "48 = { 50, 50}, % \blacktriangledown
5334 "49 = {  ,100}, % \blacktriangleright
5335 "4A = {100,  }, % \blacktriangleleft
5336 "4B = {  ,150}, % \dashrightarrow (the arrow)
5337 "4C = {150,  }, % \dashleftarrow
5338 "4D = { 50, 50}, % \vartriangle
5339 "4E = { 50, 50}, % \blacktriangle
5340 "4F = { 50, 50}, % \triangledown
5341 "50 = { 50, 50}, % \eqcirc
5342 "56 = {  ,150}, % \Rightarrow
5343 "57 = {150,  }, % \Leftarrow
5344 "58 = {100,300}, % \checkmark
5345 "5C = { 50, 50}, % \angle
5346 "5D = { 50, 50}, % \measuredangle
5347 "5E = { 50, 50}, % \sphericalangle
5348 "5F = {  , 50}, % \varpropto
5349 "60 = {100,100}, % \smallsmile
5350 "61 = {100,100}, % \smallfrown
5351 "62 = { 50,  }, % \Subset
5352 "63 = {  , 50}, % \Supset
5353 "66 = {150,150}, % \curlywedge
5354 "67 = {150,150}, % \curlyvee
5355 "68 = { 50,150}, % \leftthreetimes
5356 "69 = {100, 50}, % \rightthreetimes
5357 "6C = { 50, 50}, % \bumpeq

```

```

5358 "6D = { 50, 50}, % \Bumpeq
5359 "6E = {100, }, % \lll
5360 "6F = { ,100}, % \ggg
5361 "70 = { 50,100}, % \ulcorner
5362 "71 = {100, 50}, % \urcorner
5363 "75 = {150,200}, % \dotplus
5364 "76 = { 50,100}, % \backsim
5365 "78 = { 50,100}, % \llcorner
5366 "79 = {100, 50}, % \lrcorner
5367 "7C = {100,100}, % \intercal
5368 "7D = { 50, 50}, % \circledcirc
5369 "7E = { 50, 50}, % \circledast
5370 "7F = { 50, 50}, % \circleddash

```

Remaining slots in the source file.

```

5371 }
5372
5373 </msa>

```

Symbol font 'b'.

```

5374 <*msb>
5375 \SetProtrusion
5376 [ name = AMS-b ]
5377 { encoding = U,
5378   family = msb }
5379 {
5380   A = { 50, 50}, % \mathbb
5381   C = { 50, 50},
5382   G = { , 50},
5383   L = { , 50},
5384   P = { , 50},
5385   R = { , 50},
5386   T = { , 50},
5387   V = { 50, 50},
5388   X = { 50, 50},
5389   Y = { 50, 50},
5390 "00 = { 50, 50}, % \lvertneqq
5391 "01 = { 50, 50}, % \gvertneqq
5392 "02 = { 50, 50}, % \nleq
5393 "03 = { 50, 50}, % \ngeq
5394 "04 = {100, 50}, % \nless
5395 "05 = { 50,150}, % \ngtr
5396 "06 = {100, 50}, % \nprec
5397 "07 = { 50,150}, % \nsucc
5398 "08 = { 50, 50}, % \lneqq
5399 "09 = { 50, 50}, % \gneqq
5400 "0A = {100,100}, % \nleqslant
5401 "0B = {100,100}, % \ngeqslant
5402 "0C = {100, 50}, % \lneq
5403 "0D = { 50,100}, % \gneq
5404 "0E = {100, 50}, % \npreceq
5405 "0F = { 50,100}, % \nsucceq
5406 "10 = { 50, }, % \precnsim
5407 "11 = { 50, 50}, % \succnsim
5408 "12 = { 50, 50}, % \lnsim
5409 "13 = { 50, 50}, % \gnsim
5410 "14 = { 50, 50}, % \lneqq
5411 "15 = { 50, 50}, % \ngeqq
5412 "16 = { 50, 50}, % \precneqq
5413 "17 = { 50, 50}, % \succneqq
5414 "18 = { 50, 50}, % \preccurlyeq
5415 "19 = { 50, 50}, % \succcurlyeq
5416 "1A = { 50, 50}, % \lnapprox

```

```

5417 "1B = { 50, 50}, % \gnapprox
5418 "1C = {150,200}, % \nsim
5419 "1D = { 50, 50}, % \ncong
5420 "1E = {100,150}, % \diagup
5421 "1F = {100,150}, % \diagdown
5422 "20 = {100, 50}, % \varsubsetneq
5423 "21 = { 50,100}, % \varsupsetneq
5424 "22 = {100, 50}, % \subsetneqq
5425 "23 = { 50,100}, % \supsetneqq
5426 "24 = {100, 50}, % \subsetneqq
5427 "25 = { 50,100}, % \supsetneqq
5428 "26 = {100, 50}, % \varsubsetneqq
5429 "27 = { 50,100}, % \varsupsetneqq
5430 "28 = {100, 50}, % \subsetneq
5431 "29 = { 50,100}, % \supsetneq
5432 "2A = {100, 50}, % \subseteq
5433 "2B = { 50,100}, % \supseteq
5434 "2C = { 50,100}, % \nparallel
5435 "2D = {100,150}, % \nmid
5436 "2E = {150,150}, % \nshortmid
5437 "2F = {100,100}, % \nshortparallel
5438 "30 = { ,150}, % \nvdash
5439 "31 = { ,150}, % \nVdash
5440 "32 = { ,100}, % \nvDash
5441 "33 = { ,100}, % \nVDash
5442 "34 = { ,100}, % \ntrianglerighteq
5443 "35 = {100, }, % \ntrianglelefteq
5444 "36 = {100, }, % \ntriangleleft
5445 "37 = { ,100}, % \ntriangleright
5446 "38 = {100,200}, % \nleftarrow
5447 "39 = {100,200}, % \nrightrightarrow
5448 "3A = {100,100}, % \nLeftarrow
5449 "3B = { 50,100}, % \nrightarrow
5450 "3C = {100,100}, % \nLeftrightarrow
5451 "3D = {100,200}, % \nleftrightrightarrow
5452 "3E = { 50, 50}, % \divideontimes
5453 "3F = { 50, 50}, % \varnothing
5454 "60 = {200, }, % \Finv
5455 "61 = { , 50}, % \Game
5456 "68 = {100,100}, % \eqsim
5457 "69 = { 50, }, % \beth
5458 "6A = { 50, }, % \gimel
5459 "6B = {150, }, % \daleth
5460 "6C = {200, }, % \lessdot
5461 "6D = { ,200}, % \gtrdot
5462 "6E = {100,200}, % \ltimes
5463 "6F = {150,100}, % \rtimes
5464 "70 = { 50,100}, % \shortmid
5465 "71 = { 50, 50}, % \shortparallel
5466 "72 = {200,300}, % \smallsetminus
5467 "73 = {100,200}, % \thicksim
5468 "74 = { 50,100}, % \thickapprox
5469 "75 = { 50, 50}, % \approx
5470 "76 = { 50,100}, % \succapprox
5471 "77 = { 50, 50}, % \precapprox
5472 "78 = {100,100}, % \curvearrowleft
5473 "79 = { 50,150}, % \curvearrowright
5474 "7A = { 50,200}, % \digamma
5475 "7B = {100, 50}, % \varkappa
5476 "7F = {200, }, % \backepsilon

```

Remaining slots in the source file.

```
5477 }
```

```
5478
5479 </msb>
```

### 14.7.8 Euler

Euler Roman font (package `euler`).

```
5480 <*eur>
5481 \SetProtrusion
5482 [ name = euler ]
5483 { encoding = U,
5484   family = eur }
5485 {
5486   "01 = {100,100},
5487   "03 = {100,150},
5488   "06 = { ,100},
5489   "07 = {100,150},
5490   "08 = {100,100},
5491   "0A = {100,100},
5492   "0B = { , 50},
5493   "0C = { ,100},
5494   "0D = {100,100},
5495   "0E = { ,100},
5496   "0F = {100,100},
5497   "10 = {100,100},
5498   "13 = { ,100},
5499   "14 = { ,100},
5500   "15 = { , 50},
5501   "16 = { , 50},
5502   "17 = { 50,100},
5503   "18 = { 50,100},
5504   "1A = { , 50},
5505   "1B = { , 50},
5506   "1C = { 50,100},
5507   "1D = { 50,100},
5508   "1E = { 50,100},
5509   "1F = { 50,100},
5510   "20 = { , 50},
5511   "21 = { , 50},
5512   "22 = { 50,100},
5513   "24 = { , 50},
5514   "27 = { 50,100},
5515   1 = {100,100},
5516   7 = { 50,100},
5517   "3A = {300,500},
5518   "3B = {200,400},
5519   "3C = {200,100},
5520   "3D = {200,200},
5521   "3E = {100,200},
5522   A = { ,100},
5523   D = { , 50},
5524   J = { 50, },
5525   K = { , 50},
5526   L = { , 50},
5527   Q = { , 50},
5528   T = { 50, },
5529   X = { 50, 50},
5530   Y = { 50, },
5531   h = { , 50},
5532   k = { , 50},
5533 }
5534
```

Extended by the `eulervm` package.

```

5535 \SetProtrusion
5536 [ name = euler-vm ]
5537 { encoding = U,
5538   family = zeur }
5539 {
5540   "28 = {100,200},
5541   "29 = {100,200},
5542   "2A = {100,150},
5543   "2B = {100,150},
5544   "2C = {200,300},
5545   "2D = {200,300},
5546   "2E = { ,100},
5547   "2F = {100, },
5548   "3F = {150,150},
5549   "5B = { ,100},
5550   "5E = {100,100},
5551   "5F = {100,100},
5552   "80 = { , 50},
5553   "81 = {200,250},
5554   "82 = {100,200},
5555 }
5556
5557 <eur>

```

Euler Script font (`euca1`).

```

5558 <*eus>
5559 \SetProtrusion
5560 [ name = euscript ]
5561 { encoding = U,
5562   family = eus }
5563 {
5564   A = {100,100},
5565   B = { 50,100},
5566   C = { 50, 50},
5567   D = { 50,100},
5568   E = { 50,100},
5569   F = { 50, },
5570   G = { 50, },
5571   H = { ,100},
5572   K = { , 50},
5573   L = { ,150},
5574   M = { , 50},
5575   N = { , 50},
5576   O = { 50, 50},
5577   P = { 50, 50},
5578   T = { ,100},
5579   U = { , 50},
5580   V = { 50, 50},
5581   W = { 50, 50},
5582   X = { 50, 50},
5583   Y = { 50, },
5584   Z = { 50,100},
5585   "00 = {250,250},
5586   "18 = {200,200},
5587   "3A = {200,150},
5588   "40 = { ,100},
5589   "5E = {100,100},
5590   "5F = {100,100},
5591   "66 = { 50, },
5592   "67 = { , 50},
5593   "6E = {200,200},

```

```
5594 }
5595
5596 \SetProtrusion
5597 [ name = euscript-vm,
5598   load = euscript ]
5599 { encoding = U,
5600   family = zeus }
5601 {
5602   "01 = {600,600},
5603   "02 = {200,200},
5604   "03 = {200,200},
5605   "04 = {200,200},
5606   "05 = {150,150},
5607   "06 = {200,200},
5608   "07 = {200,200},
5609   "08 = {100,100},
5610   "09 = {100,100},
5611   "0A = {100,100},
5612   "0B = {100,100},
5613   "0C = {100,100},
5614   "0D = {100,100},
5615   "0E = {150,150},
5616   "0F = {100,100},
5617   "10 = {150,150},
5618   "11 = {100,100},
5619   "12 = {150,100},
5620   "13 = {100,150},
5621   "14 = {150,100},
5622   "15 = {100,150},
5623   "16 = {200,100},
5624   "17 = {100,200},
5625   "19 = {150,150},
5626   "1A = {150,100},
5627   "1B = {100,150},
5628   "1C = {100,100},
5629   "1D = {100,100},
5630   "1E = {250,100},
5631   "1F = {100,250},
5632   "20 = {150,200},
5633   "21 = {150,200},
5634   "22 = {150,150},
5635   "23 = {150,150},
5636   "24 = {100,200},
5637   "25 = {150,150},
5638   "26 = {150,150},
5639   "27 = {100,100},
5640   "28 = {100,100},
5641   "29 = {100,150},
5642   "2A = {100,100},
5643   "2B = {100,100},
5644   "2C = {100,100},
5645   "2D = {150,150},
5646   "2E = {150,150},
5647   "2F = {100,100},
5648   "30 = {100,100},
5649   "31 = {100,100},
5650   "32 = {100,100},
5651   "33 = {100,100},
5652   "34 = {100,100},
5653   "35 = {100,100},
5654   "3E = {150,150},
5655   "3F = {150,150},
5656   "60 = { ,200},
```

```

5657     "61 = {200,  },
5658     "62 = {100,100},
5659     "63 = {100,100},
5660     "64 = {100,100},
5661     "65 = {100,100},
5662     "68 = {300,  },
5663     "69 = {  ,300},
5664     "6C = {100,100},
5665     "6D = {100,100},
5666     "6F = {100,100},
5667     "72 = {100,100},
5668     "73 = {200,100},
5669     "76 = {  ,100},
5670     "77 = {100,  },
5671     "78 = { 50, 50},
5672     "79 = {100,100},
5673     "7A = {100,100},
5674     "7D = {150,150},
5675     "7E = {100,100},
5676     "A8 = {100,100},
5677     "A9 = {100,100},
5678     "AB = {200,200},
5679     "BA = {  ,200},
5680     "BB = {  ,200},
5681     "BD = {200,200},
5682     "DE = {200,200},
5683   }
5684
5685 </eus>

```

#### Euler Fraktur font (eufrak).

```

5686 <*euf>
5687 \SetProtrusion
5688   [ name = mathfrak ]
5689   { encoding = U,
5690     family = euf }
5691   {
5692     A = {  , 50},
5693     B = {  , 50},
5694     C = { 50, 50},
5695     D = {  , 80},
5696     E = { 50,  },
5697     G = {  , 50},
5698     L = {  , 80},
5699     O = {  , 50},
5700     T = {  , 80},
5701     X = { 80, 50},
5702     Z = { 80, 50},
5703     b = {  , 50},
5704     c = {  , 50},
5705     k = {  , 50},
5706     p = {  , 50},
5707     q = { 50,  },
5708     v = {  , 50},
5709     w = {  , 50},
5710     x = {  , 50},
5711     1 = {100,100},
5712     2 = { 80, 80},
5713     3 = { 80, 50},
5714     4 = { 80, 50},
5715     7 = { 50, 50},
5716     "12 = {500,500},
5717     "13 = {500,500},

```



```

5718      ! = { ,200},
5719      ' = {200,300},
5720      ( = {200, },
5721      ) = { ,200},
5722      * = {200,200},
5723      + = {200,250},
5724      - = {200,200},
5725      {,} = {300,300},
5726      . = {400,400},
5727      {=} = {200,200},
5728      : = { ,200},
5729      ; = { ,200},
5730      ] = { ,200},
5731      }
5732
5733 </euf>
5734 </cfg-u>

```

### 14.7.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europ), ITC Euro fonts (package euroitc) and marvosym<sup>17</sup>).

```

5735 <*cfg-e>
5736 \SetProtrusion
5737 <zpeu|euroitc> { encoding = U,
5738 <mvs> { encoding = {OT1,U},
5739 <zpeu> family = zpeu }
5740 <euroitc> family = {euroitc,euroitcs} }
5741 <mvs> family = mvs }
5742 {
5743 <zpeu> E = {50, }
5744 <euroitc> E = {100,50}
5745 <mvs> 164 = {50,50}, % \EUR
5746 <mvs> 068 = {50,-100}, % \EURdig
5747 }
5748
5749 <*zpeu|euroitc>
5750 \SetProtrusion
5751 { encoding = U,
5752 <zpeu> family = zpeu,
5753 <euroitc> family = {euroitc,euroitcs},
5754 shape = it* }
5755 {
5756 <zpeu> E = {100,-50}
5757 <euroitc> E = {100,}
5758 }
5759
5760 </zpeu|euroitc>
5761 <*zpeu>
5762 \SetProtrusion
5763 { encoding = U,
5764 family = {zpeus,eurosans} }
5765 {
5766 E = {100,50}
5767 }
5768
5769 \SetProtrusion
5770 { encoding = U,
5771 family = {zpeus,eurosans},

```

<sup>17</sup> Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

```

5772     shape    = it* }
5773     {
5774     E = {200, }
5775     }
5776
5777 </zpeu>
5778 </cfg-e>

```

## 14.8 Interword Spacing

Default unit is space.

```

5779 <*beta>
5780 <*m-t>
5781 %%% -----
5782 %%% INTERWORD SPACING SETTINGS
5783
5784 \SetExtraSpacing
5785 [ name = default ]
5786 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
5787 {

```

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

```
5788     {,} = { , -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)

```
5789     r = { , -300, 300},
```

- before or after lowercase characters with ascenders

```

5790     b = { , -200, 200},
5791     d = { , -200, 200},
5792     f = { , -200, 200},
5793     h = { , -200, 200},
5794     k = { , -200, 200},
5795     l = { , -200, 200},
5796     t = { , -200, 200},

```

- before of after lowercase characters with x-height plus descender with additional optical space, e. g., ‘v’, or ‘w’

```

5797     c = { , -100, 100},
5798     p = { , -100, 100},

```

```

5799         v = { , -100, 100},
5800         w = { , -100, 100},
5801         z = { , -100, 100},
5802         x = { , -100, 100},
5803         y = { , -100, 100}, % ?

```

- before of after lowercase characters with x-height plus descender without additional optical space

```

5804         i = { , 50, -50},
5805         m = { , 50, -50},
5806         n = { , 50, -50},
5807         u = { , 50, -50},

```

- after colon and semicolon

```

5808         : = { , 200, -200},
5809         ; = { , 200, -200},

```

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

```

5810         . = { , 250, -250},
5811         ! = { , 250, -250},
5812         ? = { , 250, -250},

```

The order has to be reversed when enlarging is needed.'

```

5813     }
5814

```

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' (`\fontdimen7`) for Computer Modern Roman is a third of `\fontdimen2`, i. e., 333.

```

5815 \SetExtraSpacing
5816 [ name = nonfrench-cmr,
5817   load = default,
5818   context = nonfrench ]
5819 { encoding = {OT1,T1,LY1,OT4,QX,T5},
5820   family = cmr }
5821 {

```

`latex.ltx` has:

```

\def\nonfrenchspacing{
\sffcode`. 3000
5822   . = {333,2000,-667},
\sffcode`? 3000
5823   ? = {333,2000,-667},
\sffcode`! 3000
5824   ! = {333,2000,-667},
\sffcode`: 2000
5825   : = {333,1000,-500},
\sffcode`; 1500
5826   ; = {   , 500,-333},
\sffcode`, 1250
5827   {,}= {   , 250,-200},
}
5828 }
5829

```

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.

```

5830 \SetExtraSpacing
5831 [ name      = nonfrench-default,
5832   load      = default,
5833   context   = nonfrench ]
5834 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
5835 {
5836   . = {240,2000,-667},
5837   ? = {240,2000,-667},
5838   ! = {240,2000,-667},
5839   : = {240,1000,-500},
5840   ; = {   , 500,-333},
5841   {,}= {   , 250,-200},
5842 }
5843

```

## 14.9 Additional Kerning

Default unit is 1em.

```

5844 %% -----
5845 %% ADDITIONAL KERNING
5846

```

A dummy list to be loaded when no context is active.

```
5847 \SetExtraKerning
5848 [ name = empty ]
5849 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
5850 { }
5851
```

### 14.9.1 French

For the French context.

```
5852 \SetExtraKerning
5853 [ name = french-default,
5854   context = french,
5855   unit = space ]
5856 { encoding = {OT1,T1,LY1} }
5857 {
5858   : = {1000,}, % = \fontdimen2
5859   ; = {500, }, % ~ \thinspace
5860   ! = {500, },
5861   ? = {500, },
5862 }
5863
```

This has the disadvantage that the word following a left guillemet will not be hyphenated. This might be fixed in pdf $\TeX$ .

```
5864 \SetExtraKerning
5865 [ name = french-guillemets,
5866   context = french-guillemets,
5867   load = french-default,
5868   unit = space ]
5869 { encoding = {OT1,T1,LY1} }
5870 {
5871   \guillemotleft = { ,800}, % = 0.8\fontdimen2
5872   \guillemotright = {800, },
5873 }
5874
```

### 14.9.2 Turkish

And for Turkish.

```
5875 \SetExtraKerning
5876 [ name = turkish,
5877   context = turkish,
5878   unit = space ]
5879 { encoding = {OT1,T1,LY1} }
5880 {
5881   : = {500, }, % ~ \thinspace
5882   ! = {500, },
5883   {=} = {500, },
5884 }
5885
```

### 14.9.3 Letterspacing

The settings with the 'letterspacing' context will be loaded whenever the command `\textls` resp. `\lsstyle` are used.

```
5886 %%% The following settings with `context=letterspacing'
5887 %%% will be loaded by \lsstyle and \textls:
```

5888

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).

```
5889 \SetExtraKerning
5890 [ name = letterspacing-all,
5891   context = letterspacing,
5892   preset = {1000,1000} ]
5893 { encoding = {U,OML,OMS,OMX} }
5894 { }
5895
```

The full stop should be spaced out less. Numbers are not spaced out, according to soul.

```
5896 \SetExtraKerning
5897 [ name = letterspacing-text,
5898   context = letterspacing,
5899   preset = {1000,1000} ]
5900 { }
5901 {
5902   . = {0, },
5903   0 = {0,0},
5904   1 = {0,0},
5905   2 = {0,0},
5906   3 = {0,0},
5907   4 = {0,0},
5908   5 = {0,0},
5909   6 = {0,0},
5910   7 = {0,0},
5911   8 = {0,0},
5912   9 = {0,0},
5913 }
5914
```

Also, quotation marks receive only half the kerning.

```
5915 \SetExtraKerning
5916 [ name = letterspacing-OT1,
5917   context = letterspacing,
5918   load = letterspacing-text,
5919   preset = {1000,1000} ]
5920 { encoding = {OT1,OT4,QX} }
5921 {
5922   \textquoteleft = {0,0}, \textquoteright = {0,0},
5923   \textquotedblleft = {0,0}, \textquotedblright = {0,0},
5924 }
5925
5926 \SetExtraKerning
5927 [ name = letterspacing-T1,
5928   context = letterspacing,
5929   load = letterspacing-text,
5930   preset = {1000,1000} ]
5931 { encoding = {T1,LV1,T5} }
5932 {
5933   \textquoteleft = {0,0}, \textquoteright = {0,0},
5934   \textquotedblleft = {0,0}, \textquotedblright = {0,0},
5935   \quotesinglbase = {0,0}, \quotedblbase = {0,0},
5936   \guilsinglleft = {0,0}, \guilsinglright = {0,0},
5937   \guillemotleft = {0,0}, \guillemotright = {0,0},
5938 }
5939
5940 /m-t
5941 /beta
```

5942 *(/config)*

## 15 Auxiliary File for Micro Fine Tuning

This file can be used to test protrusion and expansion settings.

```

5943 (*test)
5944 \documentclass{article}
5945
5946 %% Here you can specify the font you want to test, using
5947 %% the commands \fontfamily, \fontseries and \fontshape.
5948 %% Make sure to end all lines with a comment character!
5949 \newcommand*{\TestFont}{%
5950   \fontfamily{ppl}%
5951   \fontseries{b}%
5952   \fontshape{it}% sc, sl
5953 }
5954
5955 \usepackage{ifthen}
5956 \usepackage[T1]{fontenc}
5957 \usepackage[latin1]{inputenc}
5958 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
5959
5960 \pagestyle{empty}
5961 \setlength{\parindent}{0pt}
5962 \newcommand*\cruleftfill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
5963 \newcommand*\testprotrusion[2][ ]{%
5964   \ifthenelse{\equal{#1}{r}}{\cruleftfill}{\leftarrowfill} #2
5965   lorem ipsum dolor sit amet,
5966   \ifthenelse{\equal{#1}{r}}{\cruleftfill}{\leftarrowfill} #2
5967   \ifthenelse{\equal{#1}{l}}{\cruleftfill}{\rightarrowfill}
5968   you know the rest%
5969   \ifthenelse{\equal{#1}{l}}{\cruleftfill}{\rightarrowfill}
5970   \linebreak
5971   {\fontencoding{\encodingdefault}%
5972   \fontseries{\seriesdefault}%
5973   \fontshape{\shapedefault}%
5974   \selectfont
5975   Here is the beginning of a line, \dotfill and here is its end}\linebreak
5976 }
5977 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
5978 \def\stripprefix#1>{}
5979 \newcount\charcount
5980 \begin{document}
5981
5982 \microtypesetup{expansion=false}
5983
5984 {\centering The font in this document is called by:\\
5985 \texttt{\showTestFont}\par}\bigskip
5986
5987 \TestFont\selectfont
5988 This line intentionally left empty\linebreak
5989 %% A -- Z
5990 \charcount=65
5991 \loop
5992   \testprotrusion{\char\charcount}
5993   \advance\charcount 1
5994   \ifnum\charcount < 91 \repeat
5995 %% a -- z
5996 \charcount=97
5997 \loop

```

```

5998 \testprotrusion{\char\charcount}
5999 \advance\charcount 1
6000 \ifnum\charcount < 123 \repeat
6001 %% 0 -- 9
6002 \charcount=48
6003 \loop
6004 \testprotrusion{\char\charcount}
6005 \advance\charcount 1
6006 \ifnum\charcount < 58 \repeat
6007 %%
6008 \testprotrusion[r]{,}
6009 \testprotrusion[r]{.}
6010 \testprotrusion[r]{;}
6011 \testprotrusion[r]{:}
6012 \testprotrusion[r]{?}
6013 \testprotrusion[r]{!}
6014 \testprotrusion[l]{\textexclamdown}
6015 \testprotrusion[l]{\textquestiondown}
6016 \testprotrusion[r]{()}
6017 \testprotrusion[l]{()}
6018 \testprotrusion{/}
6019 \testprotrusion{\char`\}
6020 \testprotrusion{-}
6021 \testprotrusion{\textendash}
6022 \testprotrusion{\textemdash}
6023 \testprotrusion{\textquotelleft}
6024 \testprotrusion{\textquoteright}
6025 \testprotrusion{\textquotedblleft}
6026 \testprotrusion{\textquotedblright}
6027 \testprotrusion{\quotesinglbase}
6028 \testprotrusion{\quotedblbase}
6029 \testprotrusion{\guilsinglleft}
6030 \testprotrusion{\guilsinglright}
6031 \testprotrusion{\guillemotleft}
6032 \testprotrusion{\guillemotright}
6033
6034 \newpage
6035 The following displays the current font stretched by 5%,
6036 normal, and shrunk by 5%:
6037
6038 \bigskip
6039 \newlength{\MTln}
6040 \newcommand*\teststring
6041 {ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
6042 \settowidth{\MTln}{\teststring}
6043 \microtypesetup{expansion=true}
6044
6045 \parbox{1.05\MTln}{\teststring\linebreak\}
6046 \parbox{0.95\MTln}{\teststring}\par\bigskip
6047 \parbox{0.95\MTln}{\teststring}
6048
6049 \end{document}
6050 /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 . . . . .	57
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i> ) . . . . .	104	<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 . . . . .	109	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i> ) . . . . .	61
settings for Adobe Minion (contributed by <i>Harald Harders</i> ) . . . . .	110	<code>\MT@pdfTeX@no</code> : fix concerning version check (reported by <i>Harald Harders</i> ) . . . . .	26
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance . . . . .	82	<code>\MT@permute</code> : don’t use sets for empty encoding . . . . .	84
<code>\MT@declare@sets</code> : remove spaces around set name . . . . .	70	<code>\MT@pr@split</code> : fix: allow zero and negative values . . . . .	43
<code>\MT@DeclareSet@</code> : remove spaces around first argument . . . . .	70	<code>\MT@use@set</code> : remove spaces around set name . . . . .	74
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded . . . . .	56	<code>\UseMicrotypeSet</code> : remove spaces around first argument . . . . .	74

### 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 . . . . .	71
check whether only one encoding specified . . . . .	83	<code>\MT@get@listname@</code> : alternatively check for alias font name . . . . .	57
Font Sets: declare <code>cmor</code> as an alias of <code>cmr</code> . . . . .	102	<code>\MT@get@size</code> : additional magic to catch some errors hijack <code>\set@fontsize</code> instead of <code>\set@fontsize</code> . . . . .	73
new: <code>allmath</code> and <code>basicmath</code> . . . . .	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> . . . . .	33
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 <code>\globally</code> to <code>\empty</code> . . . . .	59	<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 <code>aer</code> , <code>zer</code> and <code>hfor</code> as an alias of <code>cmr</code> . . . . .	102	<code>\MT@get@codes@name</code> : fix: specifying load option does no longer require to give a name, too . . . . .	79
<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 <code>pdfcprot</code> . . . . .	36
use one instead of five counters . . . . .	34	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too . . . . .	77
Protrusion: tweak quote characters for <code>cmr</code> variants			

**Version 1.4a (2004/11/17)**

General: new option: final	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 abczz (reported by <i>Georg Verweyen</i> )	57
optimization: use less <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 (cmr, pad, ppl)	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	74
<code>\MT@checklist@family</code> : fix: don't try alias family			

**Version 1.5 (2004/12/15)**

General: defaults: step: 4 (suggested by <i>Hàn Thê Thành</i> )	90	<code>\MT@begin@catcodes</code> : reset catcode of '=' (compatibility with Turkish babel)	57
defaults: calculate step as $\min(\text{stretch}, \text{shrink})/5$	97	<code>\MT@catcodes</code> : reset catcode of '' (compatibility with chemsym)	27
defaults: turn off expansion for DVI output	96	<code>\MT@get@highlevel</code> : don't test defaults if called after begin document	71
disable automatic expansion for DVI output	97	<code>\MT@scale@factor</code> : warning for factors outside limits	44
new option: selected, by default false (suggested by <i>Hàn Thê 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 DVIoutput 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	110		
<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)	134
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	29	<code>\MT@use@set</code> : retain current set if new set is undeclared	74
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 frenchpro; problem		<code>\MT@pdf@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 \@tfor ( <i>Andreas Böhmann's</i> idea) .....	57
\SetExpansion: don't allow automatic expansion for old pdf <sub>T</sub> <sub>E</sub> <sub>X</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 .....	60
modify \showhyphens .....	98	test whether \(\encoding)\(\dots) is defined .....	60
new value for verbose option: errors .....	90	\MT@if@list@exists: don't define \MT@#1@c@name \gloally, 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>T</sub> <sub>E</sub> <sub>X</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>T</sub> <sub>E</sub> <sub>X</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. \gloally .....	39
\LoadMicrotypeFile: new command (suggested by <i>Andreas Böhmann</i> ) .....	76	\MT@test@ast: make it simpler .....	71
\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 \fontdimen 6 is defined .....	42
new option: config to load a different main configuration file .....	91	\MT@get@listname@: made recursive .....	57
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)\(\dots) 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 p <sub>x</sub> r and t <sub>x</sub> r as aliases of p <sub>p</sub> l resp. p <sub>t</sub> m	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 \@for	32
verified settings for slanted Computer Modern Roman .....	122	\MT@map@tlist@n: new macro: used instead of \@tfor	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>T</sub> <sub>E</sub> <sub>X</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 .....	64	\MT@pr@split: get character width once only .....	43
\MT@declare@sets: warning when redefining a set ..	70	\MT@scale: fix: remove spaces in non-e- <sub>T</sub> <sub>E</sub> <sub>X</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 .....	74		

### 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 .....	68
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 Thê Thành</i> ) .....	37	\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 Thê Thành</i> ) ..	107	\MT@scale@factor: generalized .....	44
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	default list name .....	79
remove superfluous test whether \pickup@font has changed .....	67	\MT@get@highlevel: no longer check whether defaults have changed .....	71
Documentation: add explanation for error message in DVI mode .....	19	\MT@ifdefined@c@T: new macros: true case only ..	29
add explanation for error message with bitmap fonts .....	19	\MT@ifint: use \pdfmatch if available .....	30
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 .....	32
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 ..	67	\MT@map@clist@n: following L <sup>A</sup> T <sub>E</sub> X <sub>3</sub> .....	32
\MT@define@set@key@: don’t expand variables immediately (requested by <i>Georg Verweyen</i> ) .....	71	\MT@permute@@@: don’t define permutations for unused encodings .....	85
\MT@get@codes@name: ‘ <i>file name</i> ’/‘ <i>line number</i> ’ as		\MT@rem@from@clist: fix .....	33
		\MT@setup@: defer setup until the end of the preamble ..	35

### Version 1.9b (2006/01/20)

General: compatibility with the listings package (reported by <i>Holger Uhr</i> ) .....	38	add samples of micro-typographic features .....	4
compatibility with the extendedchar option of the listings package .....	38	\MT@features: use throughout the package to adjust to beta-ness .....	35
register with the soul package .....	38	\MT@ifdimen: use \pdfmatch if available .....	31
Documentation: activate expansion in the distributed PDF .....	1	\MT@warn@code@too@large: fix calculation with present factor .....	45

**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 (inputenc/utf8)	62
settings for Euro symbols (Adobe, ITC, marvosym)	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	54		
<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: true</code>	88	fix: forgotten comma in <code>mt-mvs.cfg</code> ; adapt to <code>marvosym</code> 's changed encoding	145
no tracing with trace package	66	settings for Euler Roman font	141
Documentation: add hint about unknown encodings	18	<code>\DeclareCharacterInheritance</code> : new key 'inputenc' to set the input encoding	82
include LPPL	164	<code>\MT@rem@from@clist</code> : model after <code>\@removeelement</code>	33
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@setup@</code> : compatibility with the <code>combine</code> class	35
Inheritance: adapt to <code>marvosym</code> 's changed encoding	107	<code>\SetExpansion</code> : new key: <code>inputenc</code>	77
Protrusion: complete settings for Euler Fraktur and Script fonts	144	<code>\SetProtrusion</code> : new key: <code>inputenc</code>	77

**Version 2.0 ()**

General: new option: <code>babel</code> , by default false (language-dependant setup suggested by <i>Ulrich Durr</i> )	89	<code>\MT@pdfTeX@no</code> : case 6: pdf $\TeX$ 1.40	26
new option: <code>letterspacing</code> , by default 100	90	<code>\SetExtraKerning</code> : new command: additional kerning	78
<code>\DeclareMicrotypeBabelHook</code> : new command: interaction with <code>babel</code>	76	<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.

<b>Options:</b>		<code>protrusion</code>		5
<code>DVIoutput</code>	7	<code>selected</code>	7	
<code>activate</code>	5	<code>shrink</code>	7	
<code>auto</code>	6	<code>step</code>	7	
<code>config</code>	8	<code>stretch</code>	7	
<code>draft</code>	8	<code>unit</code>	6	
<code>expansion</code>	5	<code>verbose</code>	8	
<code>factor</code>	6	<b>User Commands:</b>		
<code>final</code>	8	<code>\DeclareCharacterInheritance</code>	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` ..... 589  
  
**A**  
aoposter (package) ..... 73  
activate (option) ..... 5, 88, 157  
`\add@accent` ..... 1762, 1763  
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, 76, 100, 101, 103, 154, 156, 157  
  
**C**  
`\cf@encoding` ..... 589  
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` ..... 1325, 1330, 1732  
  
**D**  
`\DeclareCharacterInheritance` ... 9, 14, 2438, 3056  
`\DeclareMicrotypeAlias` ..... 5, 15, 31, 2144, 3057  
`\DeclareMicrotypeBabelHook` ..... 16, 25, 2177  
`\DeclareMicrotypeSet` .. 2, 8, 20, 21, 32, 1863, 3048  
`\DeclareMicrotypeSet*` ..... 8, 1863  
`\DeclareMicrotypeSetDefault` ..... 4, 10, 23, 2116  
defersetup (option) ..... 37, 89, 93, 156  
`\define@newfont` ..... 1734  
`\DisableLigatures` ..... 8, 17, 24, 2101  
`\do@subst@correction` ..... 1731  
docstrip (package) ..... 24  
draft (option) ..... 8, 89, 90  
dsfont (package) ..... 42  
DVIoutput (option) ..... 7, 21, 89, 154  
  
**E**  
eco (package) ..... 16, 102  
`\efcode` ..... 1024, 1058, 1059, 1105, 1108  
eucal (package) ..... 142  
eufrak (package) ..... 144  
euler (package) ..... 64, 141, 157  
eulervm (package) ..... 16, 103, 142, 157  
euroitc (package) ..... 145  
europs (package) ..... 145  
eurosans (package) ..... 145  
expansion (option) ..... 5, 6, 10, 88, 96  
  
**F**  
`\f@encoding` ..... 589  
`\f@size` ..... 1325, 1330, 1732  
factor (option) ..... 6, 12, 22, 90, 91, 154  
fancyvrb (package) ..... 18  
final (option) ..... 8, 23, 89, 90, 154  
fix-cm (package) ..... 13  
`\font` ..... 580  
`\font@name` ..... 817, 1734, 1747, 1749, 1770, 1790  
`\fontcharwd` ..... 796  
`\fontdimen` ..... 738, 745, 816, 821, 928, 931  
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, 66, 96, 155, 156  
  
**I**  
IEEEtran (package) ..... 66  
`\iffontchar` ..... 826  
`\ifMT@auto` ..... 123, 3122, 3169  
`\ifMT@babel` ..... 123, 3251, 3274  
`\ifMT@do` ..... 608, 691, 1371  
`\ifMT@document` ..... 164, 2557  
`\ifMT@draft` ..... 123, 3036  
`\ifMT@expansion` ..... 123, 3086, 3166  
`\ifMT@inlist@` ..... 348,  
393, 466, 636, 658, 675, 684, 704, 1327,  
1404, 1415, 1503, 1754, 1778, 1841, 1848, 2162  
`\ifMT@kerning` ..... 123, 1320, 3211, 3238, 3299  
`\ifMT@noligatures` ..... 123, 1360  
`\ifMT@nonselected` ..... 1005, 1516, 1525  
`\ifMT@noreset` ..... 1581, 1588  
`\ifMT@opt@auto` ..... 2681, 3125, 3137  
`\ifMT@opt@DVI` ..... 2681, 3068  
`\ifMT@opt@expansion` ..... 2681, 3063  
`\ifMT@protrusion` ..... 123, 489, 511, 3069  
`\ifMT@selected` ..... 123, 3152, 3172  
`\ifMT@spacing` ..... 123, 3196, 3230, 3250  
ifpdf (package) ..... 7, 96  
inputenc (package) 12, 19, 21, 38, 60, 62, 64, 155, 157  
`\inputencoding` ..... 939, 942  
  
**J**  
jurabib (package) ..... 66  
  
**K**  
kerning (option) ..... 88  
keyval (package) ..... 27, 43, 70, 103  
`\knaccode` ..... 1269, 1270, 1282, 1286, 1293



- `\knbcode` ..... 1262, 1263, 1281, 1285, 1292  
`\knbscode` ..... 1155, 1156, 1181, 1186, 1194
- L**
- `ledmac` (package) ..... 20, 22, 36, 66, 155  
`ledpar` (package) ..... 20  
`\leftmarginkern` ..... 497  
`letterspacing` (option) ..... 90, 157  
`lineno` (package) ..... 36  
`listings` (package) ..... 18, 38, 57, 156  
`lmodern` (package) ..... 6, 114  
`\LoadMicrotypeFile` ..... 10, 16, 34, 2158, 3058  
`\lrcode` ..... 731, 769, 770, 952, 956  
`\lssstyle` ..... 17, 554, 575, 1318, 1355
- M**
- `marvosym` (package) .... 15, 16, 21, 24, 107, 145, 157  
`mathdesign` (package) ..... 16, 103, 156  
`memoir` (package) ..... 17, 66  
`\Microtype@Hook` ..... 92, 2900  
`\microtypecontext` ..... 12, 16,  
     1341, 1812, 3060, 3282, 3285, 3293, 3296, 3311  
`\microtypesetup` ..... 8, 11, 2904, 3059  
`minimal` (package) ..... 38  
`\MT@abbr@ex` ..... 437  
`\MT@abbr@ex@c` ..... 437  
`\MT@abbr@ex@inh` ..... 437  
`\MT@abbr@kn` ..... 444  
`\MT@abbr@kn@c` ..... 444  
`\MT@abbr@kn@inh` ..... 444  
`\MT@abbr@nl` ..... 437  
`\MT@abbr@pr` ..... 437  
`\MT@abbr@pr@c` ..... 437  
`\MT@abbr@pr@inh` ..... 437  
`\MT@abbr@sp` ..... 444  
`\MT@abbr@sp@c` ..... 444  
`\MT@abbr@sp@inh` ..... 444  
`\MT@active@features` 1772, 1775, 1787, 1794, 1829,  
     1840, 2103, 3070, 3167, 3197, 3210, 3267, 3268  
`\MT@addto@setup` 475, 532, 1730, 1812, 1813, 1925,  
     2032, 2792, 2905, 3035, 3195, 3229, 3266, 3273  
`\MT@auto` ..... 1014, 3121, 3134, 3145, 3149  
`\MT@auto@` ..... 1002, 1014, 1018, 1076  
`\MT@autofalse` ..... 125, 3132, 3144  
`\MT@autotrue` ..... 125, 2872  
`\MT@babelfalse` ..... 132, 3045  
`\MT@babeltrue` ..... 132  
`\MT@begin@catcodes` .... 564, 1405, 1406, 1429,  
     2166, 2196, 2224, 2243, 2261, 2443, 2885, 2886  
`\MT@cat` . 901, 938, 941, 943, 1089, 1551, 1722, 1725  
`\MT@catcodes` ..... 100, 119  
`\MT@char` 759, 769, 770, 775, 776, 779, 781, 796,  
     801, 826, 952, 953, 955–957, 1046, 1058,  
     1059, 1061, 1063, 1105, 1107, 1108, 1144,  
     1155, 1156, 1162, 1163, 1169, 1170, 1173,  
     1175, 1181–1183, 1185–1188, 1251, 1262,  
     1263, 1269, 1270, 1273, 1275, 1281, 1282,  
     1284–1286, 1561, 1577, 1578, 1583, 1653,  
     1655, 1657, 1683, 1689, 2503, 2504, 2509, 2512  
`\MT@char@` ..... 1561, 1567, 1572, 1577,  
     1592, 1594, 1600, 1601, 1603, 1616, 1617,  
     1620, 1621, 1624, 1625, 1629, 1631, 1656, 1697  
`\MT@charstring` ..... 1574, 1655, 1660  
`\MT@check@font` ..... 1753, 1769, 1831  
`\MT@check@font@cx` ..... 1773, 1831  
`\MT@check@range` ..... 2645, 2647  
`\MT@check@range@` ..... 2647, 2648  
`\MT@check@rlist` ..... 2587, 2638  
`\MT@check@rlist@` ..... 2638, 2639  
`\MT@checklist@` ..... 615, 628, 1365  
`\MT@checklist@family` ..... 668  
`\MT@checklist@font` ..... 647  
`\MT@checklist@size` ..... 697  
`\MT@checksetup` ..... 2915, 2926, 2933, 2954, 2976  
`\MT@clist@break` ..... 322, 642, 660, 692, 710  
`\MT@clist@function` ..... 322  
`\MT@cnt@encoding` ..... 2524, 2531, 2532  
`\MT@cnt@family` ..... 2529, 2538, 2539  
`\MT@cnt@series` ..... 2536, 2545, 2546  
`\MT@cnt@shape` ..... 2543, 2551, 2552  
`\MT@config@file` 2875, 2883, 2884, 2888, 2889, 2892  
`\MT@context` ..... 1479, 1491, 1510, 1536  
`\MT@count` ..... 273, 274, 410, 788,  
     796, 797, 802, 803, 808, 809, 813, 821, 858, 922  
`\MT@curr@file` ..... 1409, 1410,  
     1417, 1418, 2168, 2169, 2275, 2465, 2888, 3322  
`\MT@curr@list@name` ..... 819, 861, 945,  
     1072, 1391, 1549, 1692, 1700, 1705, 1712, 1718  
`\MT@curr@set@name` ..... 1893,  
     1895–1897, 1899, 1900, 1913, 1918, 1923,  
     1926, 1950, 1954, 1983, 2014, 2020, 2025,  
     2030, 2033, 2277, 2279, 2305, 2324, 2466, 2468  
`\MT@declare@char@inh` ..... 2448, 2453, 2463  
`\MT@declare@codes` ..... 2352, 2366  
`\MT@declare@sets` ..... 1880, 1886, 1894, 2105  
`\MT@DeclareSet` ..... 1867, 1868, 1870  
`\MT@DeclareSet@` ..... 1871, 1874, 1877  
`\MT@DeclareSetAndUseIt` ..... 1865, 1866, 1870  
`\MT@def@bool@opt` .... 2758, 2774, 2778, 2787, 2796  
`\MT@def@n` ..... 184, 1859, 1860, 2181,  
     2200, 2228, 2247, 2265, 2354, 2412, 2470, 2898  
`\MT@def@num@opt` ..... 2822, 2837  
`\MT@def@simple@bool@opt` ..... 2774, 2777  
`\MT@default@ex@set` ..... 2132, 3176  
`\MT@default@kn@set` ..... 2132, 3218  
`\MT@default@pr@set` ..... 2132, 3079  
`\MT@default@sp@set` ..... 2132, 3203  
`\MT@define@code@key` . 2287, 2346–2349, 2488–2490  
`\MT@define@code@key@font` ..... 2310, 2351, 2492  
`\MT@define@code@key@size` ..... 2298, 2350, 2491  
`\MT@define@context` ..... 1836, 1857  
`\MT@define@ex@c@key` ..... 2408, 2421–2423  
`\MT@define@key@unit` ..... 2386, 2405, 2406  
`\MT@define@option` ..... 2684, 2720, 2721  
`\MT@define@option@` ..... 2727, 2755, 2756  
`\MT@define@optionX` ..... 2904, 2964, 2965  
`\MT@define@optionX@` ..... 2967, 2997, 2998  
`\MT@define@set@key@` ..... 1904–1907, 1911  
`\MT@define@set@key@font` ..... 1909, 2018  
`\MT@define@set@key@size` ..... 1908, 1943  
`\MT@define@set@keys` ..... 1903, 2069, 2070  
`\MT@detokenize@c` ..... 225, 1569, 1653  
`\MT@detokenize@n` ..... 225, 1686, 1687  
`\MT@dimen@six` ..... 737, 789, 813, 858  
`\MT@diminfo` ..... 46  
`\MT@diminfo@nl` ..... 46

- \MT@do@font ..... 409, 730, 1024, 1193, 1291  
 \MT@documentfalse ..... 164  
 \MT@documenttrue ..... 164, 3270  
 \MT@dofalse .... 608, 619, 641, 663, 680, 709, 1369  
 \MT@dotrue ..... 608, 611, 638, 677, 686, 706, 1361  
 \MT@draftfalse ..... 128, 2798  
 \MT@drafttrue ..... 128, 2800  
 \MT@edef@n .....  
     185, 1515, 1540, 1852, 2090, 2094, 2136,  
     2141, 2152, 2275, 2293, 2315, 2464, 2468,  
     2479, 2610, 2618, 2704, 2706, 2741, 2743, 2828  
 \MT@encoding ..... 598,  
     1474, 1486, 1569, 1653, 1685, 1704, 1710, 1717  
 \MT@encoding@check ..... 2477, 2481  
 \MT@end@catcodes ..... 1407,  
     1446, 2174, 2201, 2229, 2248, 2266, 2458, 2887  
 \MT@error ..... 36, 469, 479, 945  
 \MT@ex@c@name ..... 998, 1000,  
     1078, 1079, 1097, 1098, 1112, 1517, 2203, 2228  
 \MT@ex@context ..... 1823, 1858  
 \MT@ex@do ..... 1001, 1036  
 \MT@ex@doc@contexts ..... 1858  
 \MT@ex@factor ..... 134, 1015, 1082  
 \MT@ex@factor@ 1015, 1026, 1029, 1048, 1049, 1076  
 \MT@ex@inh@name ..... 1060, 1061, 1063  
 \MT@ex@level ..... 134, 3168, 3170  
 \MT@ex@max ..... 149, 1051, 1052  
 \MT@ex@min ..... 149, 1054, 1055  
 \MT@ex@setname ..... 2087, 3173, 3174, 3176, 3177  
 \MT@ex@split ..... 1038, 1042  
 \MT@exp@one@n ..... 190,  
     328, 465, 586, 588, 633, 652, 1414, 1769,  
     1776, 1808, 1840, 1846, 1858, 1886, 1925,  
     2032, 2080, 2125, 2161, 2453, 3282, 3293, 3311  
 \MT@exp@string 189, 583, 740, 828, 835, 1330, 1529  
 \MT@exp@two@c ..... 191,  
     216, 220, 231, 362, 585, 1568, 1573, 1654  
 \MT@exp@two@n .... 192, 673, 682, 1323, 1917, 2024  
 \MT@expansion ..... 591, 992, 3188  
 \MT@expansionfalse ..... 124, 3041, 3064, 3162  
 \MT@expansiontrue ..... 124, 2871  
 \MT@extra@auto ..... 2203, 2428, 2434  
 \MT@extra@context 1858, 2185, 2203, 2232, 2250,  
     2315, 2319, 2320, 2323, 2362, 2439, 2584,  
     2590, 2594, 2598, 2601, 2606, 2607, 2610, 2644  
 \MT@extra@factor .... 2185, 2203, 2232, 2250, 2358  
 \MT@extra@inputenc ..... 2203,  
     2232, 2250, 2364, 2440, 2460, 2467, 2468  
 \MT@extra@load .....  
     .. 2185, 2203, 2232, 2250, 2278, 2279, 2356  
 \MT@extra@preset .... 2185, 2203, 2232, 2250, 2360  
 \MT@extra@shrink ..... 2203  
 \MT@extra@step ..... 2203  
 \MT@extra@stretch ..... 2203  
 \MT@extra@unit ..... 2185,  
     2232, 2250, 2368, 2378, 2389, 2392, 2395  
 \MT@factor@default ..... 159, 2840, 2846, 3073  
 \MT@family ..... 586, 598, 674, 1475, 2153, 2154  
 \MT@familyalias ..... 587,  
     588, 604, 681, 683, 685, 687, 1485, 1487, 2155  
 \MT@feat ..... 608,  
     742, 840, 842, 844, 845, 847, 848, 854, 856,  
     859, 867–871, 873, 875–879, 882, 883, 888,  
     892, 895, 898, 903–905, 914, 915, 920, 938,  
     941, 943, 1381, 1383, 1386, 1387, 1391–  
     1393, 1395, 1511, 1513, 1515, 1519, 1520,  
     1524, 1528, 1531, 1538, 1540, 1542, 1543,  
     1545, 1548, 1554, 1558, 1693, 1694, 1722, 1725  
 \MT@features ..... 458,  
     1858, 1880, 2069, 2074, 2119, 2366, 2448, 2896  
 \MT@features@long ..... 458, 465, 470, 1857, 2461  
 \MT@file@list ..... 1401,  
     1403, 1411, 1414, 1419, 1422, 2161, 2165  
 \MT@find@file ..... 586, 588, 1401  
 \MT@fix@font@spec ..... 1922, 1939, 2029  
 \MT@font 583, 585, 656, 659, 662, 731, 732, 738,  
     740, 745, 769, 770, 775, 776, 796, 801, 816,  
     821, 826, 828, 835, 928, 931, 952, 953, 956,  
     957, 1002, 1018, 1024, 1058, 1059, 1105,  
     1108, 1155, 1156, 1162, 1163, 1169, 1170,  
     1181–1183, 1186–1188, 1194–1196, 1262,  
     1263, 1269, 1270, 1281, 1282, 1285, 1286,  
     1292, 1293, 1372, 1458, 1529, 1729, 1746,  
     1747, 1749, 1758, 1769, 1771, 1776, 1798, 1809  
 \MT@font@list ..... 1728, 1769–1771, 1830  
 \MT@get@axis ..... 2040–2043, 2056, 2332–2335  
 \MT@get@basefamily ..... 1413, 1447  
 \MT@get@char@unit ..... 760, 890, 918, 1145, 1252  
 \MT@get@charwd ..... 794, 813, 893, 918  
 \MT@get@codes@name .. 2190, 2209, 2237, 2255, 2269  
 \MT@get@config ..... 2875  
 \MT@get@dimen@six ..... 718, 737, 1124, 1231  
 \MT@get@ex@opt ..... 995, 1009, 1076  
 \MT@get@ex@opt@ ..... 1084–1086, 1088, 1096  
 \MT@get@font ..... 2023, 2036  
 \MT@get@font@and@size ..... 2314, 2328  
 \MT@get@font@dimen ..... 815, 896  
 \MT@get@highlevel ... 1916, 1929, 2058, 2292, 2478  
 \MT@get@inh@list ..... 721, 997, 1127, 1234, 1536  
 \MT@get@listname ..... 1457, 1513, 1538  
 \MT@get@listname@ ..... 1457  
 \MT@get@opt ..... 719, 865, 1125, 1232  
 \MT@get@range ..... 1947, 1957, 2302  
 \MT@get@size .. 1964, 1969, 1978, 1997, 2050, 2343  
 \MT@get@slot 758, 1045, 1143, 1250, 1561, 2502, 2508  
 \MT@get@space@unit ..... 890,  
     982, 1153, 1160, 1167, 1214, 1218, 1260, 1267  
 \MT@get@unit ..... 898, 908  
 \MT@get@unit@ ..... 908  
 \MT@gobble@settings ..... 737  
 \MT@if@list@exists 717, 994, 1008, 1123, 1230, 1510  
 \MT@ifdefined@c@T 196, 523, 538, 778, 1060, 1172,  
     1272, 2153, 2211, 2278, 2467, 2573, 2900, 2903  
 \MT@ifdefined@c@TF ..... 196, 490, 939,  
     1338, 1514, 1539, 2583, 3076, 3173, 3200, 3215  
 \MT@ifdefined@n@T ..... 196, 630, 649,  
     670, 699, 779, 903, 937, 1061, 1091, 1173,  
     1273, 1493, 1721, 1896, 2148, 2271, 2283,  
     2532, 2539, 2546, 2552, 2584, 2598, 3019, 3309  
 \MT@ifdefined@n@TF ..... 196, 604, 613, 631,  
     650, 671, 700, 867, 875, 1078, 1097, 1363,  
     1386, 1481, 1569, 2089, 2093, 2134, 2270,  
     2559, 2617, 2633, 2703, 2740, 2897, 3280, 3291  
 \MT@ifdim .. 282, 382, 383, 387, 388, 1980, 1989,  
     2649–2651, 2653, 2654, 2661–2663, 2666, 2667  
 \MT@ifdimen ..... 261, 2009, 2377, 2394, 2860



- `\MT@ifempty` ..... 233, 767, 773, 1152, 1159, 1166, 1259, 1266, 1352, 1842, 1879, 1884, 1931, 1937, 1958, 1959, 1972, 1973, 2037, 2044, 2059, 2073, 2078, 2118, 2123, 2329, 2337, 2354, 2356, 2358, 2360, 2362, 2364, 2410, 2447, 2452, 2482, 2689, 2732, 2760, 2805, 2824, 2839, 2850, 2877, 2912, 2973  
`\MT@ifint` ..... 244, 1623, 2411, 2827, 2842  
`\MT@ifstreq` ..... 289, 1309, 1382, 1452–1454, 1512, 1589, 1807, 2370, 2376, 2391, 2393, 2692, 2694, 2697, 2700, 2735, 2737, 2763, 2764, 2808, 2809, 2814, 2853, 2859, 2914, 2921, 2925, 2932, 2975, 2981, 3253  
`\MT@in@clist` ..... 348, 465, 633, 653, 673, 682, 1323, 1403, 1414, 1769, 1776, 1840, 2161  
`\MT@in@rlist` ..... 376, 702, 1502  
`\MT@in@rlist@` ..... 376  
`\MT@in@rlist@@` ..... 376  
`\MT@in@tlist` ..... 364, 1846  
`\MT@in@tlist@` ..... 364  
`\MT@increment` ..... 410, 2531, 2538, 2545, 2551  
`\MT@info` ..... 36, 50, 583, 2091, 2917, 2923, 2929, 2936, 2978, 2983, 3007, 3011  
`\MT@info@missing@char` ..... 797, 803, 824  
`\MT@info@nl` ..... 36, 47, 51, 491, 582, 826, 833, 2804, 2884, 3067, 3072, 3077, 3080, 3084, 3169, 3174, 3177, 3189, 3199, 3201, 3204, 3208, 3214, 3216, 3219, 3222, 3234, 3242, 3276  
`\MT@inh@do` ..... 1553, 2493  
`\MT@inh@feat` ..... 2441, 2447, 2450, 2462  
`\MT@inh@name` ..... 1548, 1549, 2511, 2518  
`\MT@inh@split` ..... 2495, 2499  
`\MT@inlist@false` ..... 348, 352, 365, 377, 1784  
`\MT@inlist@true` ..... 348, 354, 372, 384, 389, 1784  
`\MT@is@active` ..... 1568, 1636  
`\MT@is@char` ..... 1573, 1654, 1660  
`\MT@is@composite` ..... 1571, 1681  
`\MT@is@feature` ..... 464, 1885, 2079, 2124  
`\MT@is@letter` ..... 1566, 1590, 1657, 1689  
`\MT@is@number` ..... 1606, 1611  
`\MT@is@symbol` ..... 1570, 1652  
`\MT@iterate` ..... 398  
`\MT@kerning` ..... 594, 1224, 3245  
`\MT@kerning@false` ..... 131, 3044, 3244  
`\MT@kerning@true` ..... 131, 1334  
`\MT@kn@cc@name` ..... 1235, 1237, 1300, 2250, 2265  
`\MT@kn@context` ..... 1309, 1824, 1858  
`\MT@kn@do` ..... 1238, 1241  
`\MT@kn@doc@contexts` ..... 1858  
`\MT@kn@factor` ..... 134  
`\MT@kn@factor@` ..... 867, 1310, 1312  
`\MT@kn@inh@name` ..... 1272, 1273, 1275  
`\MT@kn@max` ..... 149  
`\MT@kn@min` ..... 149  
`\MT@kn@setname` ..... 1326, 1329, 1337, 3215, 3216, 3218, 3219  
`\MT@kn@split` ..... 1243, 1247  
`\MT@kn@split@val` ..... 1253, 1257  
`\MT@kn@unit` ..... 134  
`\MT@kn@unit@` ..... 875, 1303  
`\MT@led@kern` ..... 487  
`\MT@led@unhbox@line` ..... 487  
`\MT@ledmac@setup` ..... 487, 534  
`\MT@let@cn` ..... 187, 188, 605, 724, 1000, 1079, 1130, 1237, 1381, 1392, 1507, 1511, 1545, 1548, 2001, 2277, 2466, 2825  
`\MT@let@enc` 186, 905, 1524, 1554, 1558, 1779, 1830, 1899, 1913, 2020, 2188, 2207, 2218, 2235, 2253, 2279, 2619, 2631, 2698, 2701, 2928, 2935  
`\MT@let@nn` ..... 188, 868, 873, 876, 888, 1098, 1101, 1845, 2284  
`\MT@letterspacing` ..... 134, 1339, 3224, 3225  
`\MT@letterspacing@` ..... 1311, 1338, 1339, 1351  
`\MT@letterspacing@default` ..... 163, 3225  
`\MT@listname` 1457, 1504, 1507, 1514, 1515, 1539, 1540  
`\MT@load@list` ..... 722, 998, 1128, 1235, 1379  
`\MT@loop` ..... 398, 405, 2630  
`\MT@lower` ..... 1951, 1957, 2305  
`\MT@make@string` ..... 436, 1941, 2147, 2160  
`\MT@map@clist@` ..... 322  
`\MT@map@clist@c` ..... 322, 1775, 1787, 1794, 1829, 1857, 1880, 1882, 2069, 2074, 2076, 2119, 2121, 2366, 2448, 2450, 2461, 2896, 3017  
`\MT@map@clist@n` ..... 322, 612, 1362, 1858, 1914, 1945, 2021, 2179, 2188, 2191, 2206, 2210, 2235, 2238, 2253, 2256, 2290, 2300, 2312, 2505, 2687, 2730, 2909, 2970  
`\MT@map@tlist@` ..... 338  
`\MT@map@tlist@c` ..... 338, 367, 378, 780, 1062, 1174, 1274, 1461, 1800, 2585, 2643  
`\MT@map@tlist@n` ..... 338, 1441, 2775, 2835  
`\MT@maybe@do` ..... 608, 715, 992, 1118, 1225  
`\MT@MT` ..... 35, 36, 37, 40, 42, 43, 95, 480, 482, 2878, 3007, 3011, 3068, 3257  
`\MT@next@listname` ..... 1464, 1473  
`\MT@next@listname@` ..... 1482, 1494, 1500  
`\MT@nl@setname` ..... 2106  
`\MT@noligatures` ..... 596, 1358  
`\MT@noligatures@false` ..... 127  
`\MT@noligatures@true` ..... 127, 2104  
`\MT@nonselected@false` ..... 1005, 1019  
`\MT@nonselected@true` ..... 1005, 1007  
`\MT@noreset@false` ..... 1588, 1589, 1596, 1605, 1626  
`\MT@noreset@true` ..... 1564, 1588, 1614  
`\MT@old@cmd` ..... 26, 31–34  
`\MT@opt@autofalse` ..... 2682  
`\MT@opt@autotruer` ..... 2682  
`\MT@opt@DVIfalse` ..... 2683  
`\MT@opt@DVIttrue` ..... 2683, 2780, 2783  
`\MT@opt@expansion@false` ..... 2681  
`\MT@opt@expansion@true` ..... 2681  
`\MT@orig@add@accent` ..... 1762  
`\MT@orig@foreign@language` ..... 3288, 3290  
`\MT@orig@pickup@font` ..... 552, 1734, 1764  
`\MT@orig@select@language` ..... 3277, 3279  
`\MT@pdfcprot@error` ..... 478, 533  
`\MT@pdf@tex@no` ..... 53, 87, 91, 178, 3148  
`\MT@permute` ... 2195, 2223, 2242, 2260, 2473, 2523  
`\MT@permute@` ..... 2523  
`\MT@permute@@` ..... 2523  
`\MT@permute@@@` ..... 2523  
`\MT@permute@@@@` ..... 2550, 2555  
`\MT@permute@@@@@` ..... 2563, 2565  
`\MT@permute@define` ..... 2556, 2566–2568, 2615  
`\MT@permute@reset` ..... 2526, 2621

- \MT@permute@reset@ ..... 2622–2625, 2628  
 \MT@permutelist .. 2185, 2221, 2240, 2258, 2315,  
     2316, 2323, 2471, 2579, 2580, 2584, 2590,  
     2594, 2598–2601, 2610, 2611, 2644, 2675, 2676  
 \MT@pickupfont ..... 1761, 1766  
 \MT@pr@cname ..... 722, 724, 962, 2185, 2200  
 \MT@pr@context ..... 1823, 1858  
 \MT@pr@do ..... 725, 749  
 \MT@pr@doc@contexts ..... 1858  
 \MT@pr@factor ..... 134, 2843, 3073, 3074  
 \MT@pr@factor@ ..... 867  
 \MT@pr@inh@name ..... 778, 779, 781  
 \MT@pr@level ..... 134, 3071, 3072  
 \MT@pr@max ..... 149  
 \MT@pr@min ..... 149  
 \MT@pr@setname ..... 2087, 3076, 3077, 3079, 3080  
 \MT@pr@split ..... 751, 755  
 \MT@pr@split@val ..... 761, 765  
 \MT@pr@unit ..... 134, 2861, 3075  
 \MT@pr@unit@ ..... 875, 965  
 \MT@preset@aux ..... 967,  
     969, 971, 972, 975, 1305, 1307, 1314, 1315  
 \MT@preset@aux@factor .. 967, 975, 1208–1210, 1305  
 \MT@preset@aux@space ..... 969, 975, 1212, 1307  
 \MT@preset@ex ..... 1092, 1111  
 \MT@preset@kn ..... 1298  
 \MT@preset@kn@ ..... 1298  
 \MT@preset@pr ..... 960  
 \MT@preset@pr@ ..... 960  
 \MT@preset@sp ..... 1201  
 \MT@preset@sp@ ..... 1201  
 \MT@ProcessOptionsWithKV ..... 3014, 3034  
 \MT@protrusion ..... 590, 715, 3083  
 \MT@protrusionfalse ..... 123, 3040  
 \MT@protrusiontrue ..... 123, 2868  
 \MT@rba@expansion ..... 452  
 \MT@rba@kerning ..... 452  
 \MT@rba@protrusion ..... 452  
 \MT@rba@spacing ..... 452  
 \MT@register@font ..... 1756, 1771, 1832  
 \MT@register@font@cx ..... 1793, 1832  
 \MT@register@subst@font ..... 1750, 1770, 1833  
 \MT@register@subst@font@cx ..... 1786, 1833  
 \MT@rem@from@clist ..... 359, 1808  
 \MT@rem@from@lists ..... 1802, 1806  
 \MT@repeat ..... 398, 407, 2636  
 \MT@requires@etex 165, 196, 225, 411, 420, 794, 824  
 \MT@requires@pdftex ..... 177, 244,  
     261, 289, 487, 786, 1027, 1117, 1224, 1318,  
     1358, 2101, 2427, 2433, 2870, 3123, 3194, 3248  
 \MT@reset@context ..... 1818, 1821  
 \MT@reset@ef@codes ..... 996, 1017, 1027, 1093  
 \MT@reset@ef@codes@ ..... 1022, 1030, 1034  
 \MT@reset@kn@codes ..... 1233, 1239, 1289  
 \MT@reset@kn@codes@ ..... 1289  
 \MT@reset@pr@codes ..... 720, 726, 735  
 \MT@reset@pr@codes@ ..... 735  
 \MT@reset@sp@codes ..... 1126, 1132, 1191  
 \MT@reset@sp@codes@ ..... 1191  
 \MT@restore@catcodes ..... 112, 120  
 \MT@restore@p@h ..... 520, 528, 529, 550, 560  
 \MT@saved@setupfont ..... 3005, 3012  
 \MT@scale ..... 420, 789, 841, 855, 858, 1049, 1311  
 \MT@scale@factor ..... 791, 810, 839, 977, 1113  
 \MT@scale@to@em ..... 768, 774, 786, 807,  
     983, 1154, 1161, 1168, 1215, 1219, 1261, 1268  
 \MT@selectedfalse ..... 126  
 \MT@selectedtrue ..... 126  
 \MT@series ..... 598, 1476, 1488  
 \MT@set@all@ex ..... 1022, 1114  
 \MT@set@all@kn ..... 1289, 1316  
 \MT@set@all@pr ..... 728, 735, 973  
 \MT@set@all@sp ..... 1191, 1222  
 \MT@set@default@set ..... 2119, 2125, 2132  
 \MT@set@ex@codes ..... 1021, 3153, 3155  
 \MT@set@ex@codes@n ..... 1005, 1021, 3155  
 \MT@set@ex@codes@s ..... 993, 3153  
 \MT@set@ex@heirs ..... 1064, 1104  
 \MT@set@ex@list ..... 2225, 2227  
 \MT@set@inh@list ..... 2444, 2446  
 \MT@set@inputenc ..... 902, 936, 1090, 1552  
 \MT@set@kn@codes ..... 1229  
 \MT@set@kn@heirs ..... 1276, 1280  
 \MT@set@kn@list ..... 2262, 2264  
 \MT@set@listname .....  
     ..... 723, 866, 999, 1077, 1129, 1236, 1692  
 \MT@set@opt ..... 2191, 2210, 2238, 2256, 2282  
 \MT@set@pr@codes ..... 716  
 \MT@set@pr@heirs ..... 782, 951  
 \MT@set@pr@list ..... 2197, 2199  
 \MT@set@sp@codes ..... 1122  
 \MT@set@sp@heirs ..... 1176, 1180  
 \MT@set@sp@list ..... 2244, 2246  
 \MT@setup@ ..... 474, 475, 476, 2790, 2791  
 \MT@setup@contexts ..... 1815, 1828  
 \MT@setupfont .. 581, 1755, 3005, 3008, 3012, 3047  
 \MT@setupfont@hook .....  
     ..... 521, 535, 537, 542, 550, 560, 569, 584  
 \MT@shape ..... 598, 1477, 1489  
 \MT@shrink ..... 134, 1012, 3090,  
     3092, 3094, 3098, 3099, 3102, 3106, 3158, 3171  
 \MT@shrink@ ..... 1002, 1012, 1018, 1076  
 \MT@shrink@default ..... 160, 3094  
 \MT@size ..... 383, 387, 388, 598, 705, 708  
 \MT@size@name ..... 376, 1504  
 \MT@sp@cname ..... 1128, 1130, 1203, 2232, 2247  
 \MT@sp@context ..... 1824, 1858, 3253  
 \MT@sp@do ..... 1131, 1134  
 \MT@sp@doc@contexts ..... 1858  
 \MT@sp@factor ..... 134  
 \MT@sp@factor@ ..... 867  
 \MT@sp@inh@name ..... 1172, 1173, 1175  
 \MT@sp@max ..... 149  
 \MT@sp@min ..... 149  
 \MT@sp@setname ..... 3200, 3201, 3203, 3204  
 \MT@sp@split ..... 1136, 1140  
 \MT@sp@split@val ..... 1146, 1150  
 \MT@sp@unit ..... 134  
 \MT@sp@unit@ ..... 875, 1206  
 \MT@spacing ..... 593, 1117, 3207, 3237  
 \MT@spacingfalse ..... 130, 3043, 3236  
 \MT@spacingtrue ..... 130  
 \MT@split@name ..... 585, 598  
 \MT@split@val ..... 1150  
 \MT@step ..... 134, 1013, 3097, 3113, 3120, 3172  
 \MT@step@ ..... 1002, 1013, 1018, 1076

- `\MT@step@default` ..... 160  
`\MT@stretch` ..... 134, 1011, 3087, 3088,  
3091, 3092, 3098, 3100, 3105, 3108, 3157, 3171  
`\MT@stretch@` ..... 1002, 1011, 1018, 1076  
`\MT@stretch@default` ..... 160, 3088  
`\MT@temp` ..... 2039, 2067, 2331  
`\MT@tempencoding` ..... 2558, 2559, 2569, 2576  
`\MT@tempfamily` ..... 2570  
`\MT@tempseries` ..... 2571  
`\MT@tempshape` ..... 2572  
`\MT@tempsize` ..... 2573, 2583, 2586, 2591, 2626  
`\MT@test@ast` ..... 1930, 1935  
`\MT@testrest` ..... 1588  
`\MT@tlist@break` ..... 338, 373, 395, 1466, 2678  
`\MT@toks` ..... 122, 826, 834, 861,  
1072, 1565, 1594, 1601, 1646, 1697, 1709, 1716  
`\MT@try@order` ..... 1461, 1469  
`\MT@undefined@char` ..... 1641, 1643, 1651  
`\MT@upper` ..... 1951, 1957, 2305  
`\MT@use@set` ..... 2074, 2080, 2087  
`\MT@val` ..... 1842–1844,  
1847, 1849, 1852, 1915, 1919, 1930, 1932,  
1946–1948, 1960, 1963, 1965, 1968, 1970,  
1971, 1977, 1979, 1980, 1982, 1985, 1987,  
1998, 2001, 2005, 2009–2011, 2013, 2015,  
2022, 2023, 2026, 2046, 2048, 2051–2053,  
2057, 2059, 2061, 2063, 2291, 2293, 2301–  
2303, 2313, 2314, 2318, 2324, 2339, 2341,  
2476, 2477, 2479, 2483, 2504, 2511, 2517,  
2518, 2688, 2689, 2692, 2694, 2697, 2700,  
2703, 2704, 2709, 2731, 2732, 2735, 2737,  
2740, 2741, 2746, 2910, 2912, 2914, 2921,  
2925, 2932, 2940, 2971, 2973, 2975, 2981, 2986  
`\MT@vinfo` ..... 36, 47, 582, 625,  
870, 879, 883, 920, 1080, 1099, 1373, 1387,  
1410, 1418, 1421, 1517, 1519, 1526, 1822,  
1844, 2163, 2169, 2804, 2813, 3281, 3292, 3310  
`\MT@warn@ascii` ..... 1603, 1696  
`\MT@warn@axis@empty` ..... 2045, 2060, 2065, 2338  
`\MT@warn@code@too@large` ..... 845, 848, 852  
`\MT@warn@err` ..... 36, 2810, 2811  
`\MT@warn@ex@too@large` ..... 1052, 1055, 1070  
`\MT@warn@maybe@inputenc` ..... 1710, 1717, 1720  
`\MT@warn@number@too@large` ..... 1630, 1702  
`\MT@warn@preset@tewidth` ..... 966, 986, 1207, 1304  
`\MT@warn@rest` ..... 1582, 1707  
`\MT@warn@unknown` ..... 1579, 1714  
`\MT@warning` ..... 27, 36,  
1345, 1383, 1395, 1528, 1897, 1981, 2013,  
2066, 2096, 2110, 2138, 2149, 2171, 2272,  
2371, 2380, 2397, 2414, 2430, 2484, 2577,  
2599, 2675, 2810, 2854, 2900, 2939, 2958, 2985  
`\MT@warning@nl` ..... 36, 90, 503, 512, 544, 739,  
817, 859, 913, 987, 1071, 1697, 1703, 1708,  
1715, 1736, 2213, 2708, 2745, 2765, 2811,  
2815, 2829, 2844, 2863, 2891, 3037, 3115,  
3126, 3138, 3159, 3231, 3239, 3254, 3306, 3315  
`\MT@while@num` ..... 403, 409, 565–567  
`\MT@with@package` ..... 477, 486,  
525, 529, 533, 534, 540, 559, 563, 574, 1743  
`\MT@xadd` ..... 308, 1328, 1411, 1419,  
1422, 1788, 1796, 1849, 1949, 2165, 2304, 2510  
`\MT@xaddb` ..... 315, 2322, 2589
- N**
- `\normalfont` ..... 580, 3181, 3185  
`\nullfont` ..... 580
- P**
- `\pdfadjustinterwordglue` ..... 2997, 3198  
`\pdfadjustspacing` ..... 2965, 3168, 3184  
`\pdfappendkern` ..... 1335, 2999, 3213  
`pdfcpot` (package) ..... 5, 19, 20, 36, 43, 65, 153  
`\pdffontexpand` ..... 1002, 1018  
`\pdfmatch` ..... 246, 263  
`\pdfnoligatures` ..... 1372  
`\pdfoutput` ..... 57,  
2780, 2781, 2783, 2784, 2869, 3062, 3067, 3124  
`\pdfprependkern` ..... 1336, 2998, 3212  
`\pdfprotrudechars` ..... 2964, 3071  
`\pdfstrcmp` ..... 291  
`\pdfTeXrevision` ..... 56, 61, 70, 72  
`\pdfTeXversion` ..... 55, 59–61, 63, 65, 67, 69, 77  
`\pickup@font` ..... 552, 1735, 1737, 1742–1744, 1761, 1764, 1766  
`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**
- `relsize` (package) ..... 73  
`\rightmargin@kern` ..... 496  
`\rPCODE` ..... 732, 775, 776, 953, 957
- S**
- `selected` (option) ..... 7, 13, 14, 22, 48, 59, 89, 154  
`\selectfont` ..... 1817, 1826, 3321  
`\set@fontsize` ..... 2005  
`\SetExpansion` ..... 7, 13, 2203, 3051  
`\SetExtraKerning` ..... 15, 2250, 3054  
`\SetExtraSpacing` ..... 14, 2232, 3053  
`\SetProtrusion` ..... 6, 11, 2185, 3050  
`\sfcode` ..... 3252  
`\shbscode` ..... 1169, 1170, 1183, 1188, 1196  
`\showhyphens` ..... 3179, 3183  
`shrink` (option) ..... 7, 14, 22, 90, 97, 156  
`slantsc` (package) ..... 127  
`soul` (package) ..... 38, 150, 156  
`spacing` (option) ..... 88  
`\stbcode` ..... 1162, 1163, 1182, 1187, 1195  
`step` (option) ..... 7, 22, 90, 97, 154, 156  
`stretch` (option) ..... 7, 14, 18, 22, 90, 97
- T**
- `tex4ht` (package) ..... 37, 157  
`\textls` ..... 18, 555, 576, 1351  
`tipa` (package) ..... 66  
`trace` (package) ..... 66, 157  
`\tracingmicrotype` ..... 46  
`txfonts` (package) ..... 16, 102
- U**
- `ucs` (package) ..... 19, 62

unit (option) . . . . .	6, 22, 155, 156				
\UseMicrotypeSet . . . . .	3, 10, 22, 33, 1875, 2071, 3049	xkeyval (package) . . . . .			80
				<b>X</b>	
				<b>V</b>	
verbose (option) . . . . .	8, 22, 90, 155	zefonts (package) . . . . .			16, 102
				<b>Z</b>	

## 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 ‘c<sub>f</sub>gguide.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 L<sup>P</sup>PL 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.