



The microtype package

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

R Schlicht w.m.l@gmx.net

v2.0 — 2007/01/21

Abstract

The microtype package provides an interface to the micro-typographic extensions of pdf \TeX : most prominently, character protrusion and font expansion, furthermore the adjustment of interword spacing and additional kerning, as well as hyphenatable letterspacing (tracking) and the possibility to disable all ligatures of a font. It allows to apply these features to customisable sets of fonts, and to configure all micro-typographic aspects of the fonts in a straight-forward and flexible way. Settings for various fonts are provided.¹

Note that font expansion and character protrusion will only work with pdf \TeX , at least version 0.14f. Automatic font expansion requires version 1.20 or newer. Disabling ligatures requires pdf \TeX 1.30, letterspacing and the adjustment of interword spacing and of kerning requires version 1.40. The package will by default enable protrusion and expansion if they can safely be assumed to work.

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

This package is copyright © 2004–2007 R Schlicht. It may be distributed and/or modified under the conditions of the [L^AT_EX 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’.

¹ Currently, this package provides protrusion 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

1	Micro-Typography with pdf\TeX	4
2	Invoking the Package	5
3	Options	6
3.1	Micro-Typographic Options	6
3.2	Options for Character Protrusion	7
3.3	Options for Font Expansion	7
3.4	Option for Tracking/Letterspacing	8
3.5	Miscellaneous Options	8
3.6	Changing Options Later	9
4	Selecting Fonts for Micro-Typography	9
5	Micro Fine Tuning	12
5.1	Character Protrusion	12
5.2	Font Expansion	14
5.3	Tracking	15
5.4	Interword Spacing	16
5.5	Additional Kerning	17
5.6	Character Inheritance	18
5.7	Configuration Files	18
6	Context-sensitive Setup	20
7	Letterspacing revisited	21
8	Disabling Ligatures	22
9	Hints and Caveats	22
10	Contributions	24
11	Acknowledgments	25
12	References	25
13	Short History	26
14	Implementation	29
14.1	Preliminaries	29
	[Debugging – 31] [Requirements – 32] [Declarations – 34] [Auxiliary Macros – 35] [Compatibility – 42]	
14.2	Font Setup	46
	[Protrusion – 49] [Expansion – 55] [Interword Spacing (Glue) – 58] [Additional Kerning – 60] [Tracking – 62] [Disabling Ligatures – 65] [Loading the Configuration – 65] [Translating Characters into Slots – 69] [Hook into \LaTeX 's font selection – 74] [Context-sensitive Setup – 77]	

14.3 Configuration	79
[Font Sets – 79] [Interaction with babel – 86] [Fine Tuning – 86] [Character Inheritance – 91] [Permutation – 93]	
14.4 Package Options	96
[Declaring the Options – 96] [Reading the Configuration File – 100] [Hook for Other Packages – 101] [Changing Options Later – 102] [Processing the Options – 104]	
15 Configuration Files	111
15.1 Font Sets	111
15.2 Font Aliases	112
15.3 Interaction with babel	113
15.4 Note on Admissible Characters	113
15.5 Character Inheritance	113
[OT1 – 114] [T1 – 114] [LY1 – 114] [OT4 – 115] [QX – 116] [T5 – 116]	
[Euro symbols – 117]	
15.6 Font Expansion	117
15.7 Character Protrusion	118
[Default – 119] [Italics – 126] [Small Caps – 134] [Italic Small Caps – 137]	
[textcomp – 138] [Computer Modern math – 143] [AMS symbols – 147]	
[Euler – 150] [Euro symbols – 155]	
15.8 Interword Spacing	155
[Nonfrenchspacing – 157]	
15.9 Additional Kerning	158
[French – 158] [Turkish – 159]	
16 Auxiliary File for Micro Fine Tuning	159
A Change History	162
B Index	167
C The L^AT_EX Project Public License	174

List of Tables

1 Predefined font sets	11
2 Fonts with tailored protrusion settings	19
3 Order for matching font attributes	67

1 Micro-Typography with pdf_T_EX

pdf_T_EX, the _T_EX extension written by Hàn Thế Thành, introduces a number of micro-typographic features that make it the tool of choice not only for the creation of electronic documents but also of works of outstanding time-honoured typography: most prominently, *character protrusion* (also known as margin kerning) and *font expansion*. Quoting Hàn Thế Thành's thesis:

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

Protrusion	off
Expansion	off

Both features are enabled throughout this document.

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

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

Both these features have been lacking a simple \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 utilise, since it required that the font metrics are available for all levels of expansion. Therefore, anybody who wanted to make use of this feature had to create multiple instances of the fonts in advance. Shell scripts to partly relieve the user from this burden were available – however, it remained a cumbersome task. Furthermore, all fonts were still being physically created, thus wasting compilation time and disk space.

In the summer of 2004, Hàn Thế Thành implemented a feature that has proven as a major facilitation for _T_EX and \LaTeX users: font expansion can now take place automatically. That is, pdf_T_EX no longer needs the expanded font metrics but will calculate them at run-time and completely in memory.

After this great leap in usability had been taken, the development did not stop. On the contrary, pdf_T_EX was extended with even more features: version 1.30 introduced the possibility to *disable all ligatures*, version 1.40 a robust *letterspacing* command, the *adjustment of interword spacing* and the possibility to specify *additional character kerning*.

Robust and hyphenatable *letterspacing* (tracking) has always been extremely difficult to achieve in _T_EX. Although the `soul` package undertook great efforts in making this possible, it could still fail in certain circumstances; even to adjust the tracking of a font throughout the document remained impossible. Employing pdf_T_EX's new extension, this doesn't pose a problem any longer. The microtype

package provides the possibility to change the tracking of customisable sets of fonts, e. g., small capitals. It also introduces two new commands `\textls` and `\lststyle` for ad-hoc letterspacing, which can be used like the normal text commands.

Adjustment of interword spacing is based upon the idea that in order to achieve a uniform greyness of the text, the space between words should also depend on the surrounding characters. For example, if a words ends with an ‘r’, the following space should be a tiny bit smaller than that following an, say, ‘m’. You can think of this concept as an extension to \TeX ’s ‘space factors’. However, while space factors will influence all three parameters of interword space (or glue) by the same amount – the kerning, the maximum amount that the space may be stretched and the maximum amount that it may be shrunk –, \pdf\TeX provides the possibility to modify these parameters independently from one another. Furthermore, the values may be set differently for each font. And, probably most importantly, the parameters may not only be increased but also decreased. This feature may enhance the appearance of paragraphs even more. Emphasis in the last sentence is on the word ‘may’: this extension is still highly experimental – in particular, only ending characters will currently have an influence on the interword space. Also, the settings that are shipped with `microtype` are but a first approximation, and I would welcome corrections and improvements very much. I suggest reading the reasoning behind the settings in section 15.8.

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

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

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

2 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 \pdfTeX version).

3.1 Micro-Typographic Options

protrusion true, false, compatibility, nocompatibility, ** *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 \pdfTeX cannot expand the fonts automatically, that is, if it is either too old (versions before 1.20) or if the output mode is DVI (see section 3.5). In other words, `microtype` will try to apply as much micro-typography as can safely be expected to work under the respective conditions (and it is usually not necessary to load the package with different options for PDF resp. DVI mode).

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

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

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

```
\usepackage{microtype}
```

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.

tracking true, false, ** false

spacing The new extensions of tracking, interword spacing, and additional kerning do not have a compatibility level. Therefore, they can only be switched on or off, or they may be activated by passing a set name to the option. By default, neither feature is enabled.

kerning

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

3.2 Options for Character Protrusion

factor *<integer>* 1000

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

unit character, *<dimension>* character

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

3.3 Options for Font Expansion

auto true, false *true

As noted in chapter 1, the expanded versions of the fonts may be calculated automatically. This option is true by default provided that pdf_TE_X'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 *±<expansion value>*, e. g., *cmr12+10*, as described in the pdf_TE_X manual). 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 encoding², you should either install the cm-super fonts or use the Latin Modern fonts (package `lmodern`).

stretch *<integer>* 20

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

step *<integer>* min(stretch,shrink)/5

Font expansion will be applied in discrete steps. For example, if step is set to 4 (which it is by default), pdf_TE_X 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:

² 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’.

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

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

```
\usepackage{microtype}
```

selected true, false false

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

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

3.4 Option for Tracking/Letterspacing

letterspace *<integer>* 100

This option changes the default amount for tracking (see section 5.3) resp. letterspacing (see section 7). The amount is specified in thousandth of 1em; admissible values are in the range of –1000 to +1000.

3.5 Miscellaneous Options

DVIoutput true, false false

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

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

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

draft true, false false

final If the draft option is passed to the package, *all micro-typographic extensions will be disabled*, which may lead to different line, and hence page, breaks. The draft and final options may also be inherited from the class options; of course, you can override them in the package options. E. g., if you are using the class option draft to show any overfull boxes, you should load microtype with the final option.

³ Recent _T_EX systems are using pdf_T_EX as the default engine even for DVI output.

verbose true, false, errors false

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

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

babel true, false false

Loading the package with the babel option will adjust the typesetting according to the respective selected language. Read section 6 for further information.

config <file name> microtype

Various settings for this package will be loaded from a main configuration file, by default microtype.cfg (see section 5.7). You can have a different configuration file loaded instead by specifying its name *without the extension*, e. g., config=mycrottype.

3.6 Changing Options Later

\microtypesetup {<key = value list>}

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

In the document body, this command may be used to change the general settings of the micro-typographic extensions. It then accepts the keys: **expansion**, **protrusion** and **activate**, which in turn may receive the values true, false, compatibility or nocompatibility (but not the name of a font set). Since there is no compatibility level for **tracking**, **spacing** and **kerning**, only the values true and false are admissible here. Using this command, you could for instance temporarily disable font expansion by saying:

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

4 Selecting Fonts for Micro-Typography

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

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

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

The set of fonts is specified by assigning values to the NFSS font attributes: encoding, family, series, shape and size (cf. [L^AT_EX 2_ε font selection](#)). Let's start with an example. This package defines a font set called 'basictext' in the main configuration file as follows:

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

If you now call

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

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

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

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

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

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

Sizes may be either specified as a dimension ('10' or '10pt'), or as a size selection command *without* the backslash. You may also specify ranges (e. g., 'small-Large'); 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 (allmath)	Text encodings, TS1 (OML, OMS, U)	–	–	–	–
basictext (basicmath)	Text encodings (OML, OMS)	$\rm*$, $\sf*$	$\md*$	–	\normalsize , \footnotesize , \small , \large
smallcaps	Text encodings	–	–	sc	–
footnotesize (scriptsize)	Text encodings, TS1	–	–	–	\small (\footnotesize)
normalfont	$\encoding*$	$\family*$	$\series*$	$\shape*$	\normalsize

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

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

```

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

```

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

This command activates a font set previously declared by `\DeclareMicrotypeSet`. Using the optional argument, you can limit the application of the set to one or more features. This command only has an effect if the feature has been activated in the package options.

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

If 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 and additional kerning, the ‘basictext’ set for font expansion and interword spacing.

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

5 Micro Fine Tuning

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

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

The set of fonts to which the settings should apply is declared using the same syntax of $\langle font axis \rangle = \langle value list \rangle$ 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.⁴ The encoding must always match.

5.1 Character Protrusion

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

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

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

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

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

The $\langle characters \rangle$ 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^AT_EX’s traditional 7-bit notation: both ‘\’A and ‘Ä’ are valid, provided the character is actually declared in both the input and the font encoding. You also have the possibility to declare lists of characters that should inherit protrusion or expansion factors (see section 5.6).

⁴ For the interested, table 3 on page 67 presents the exact order.

The *⟨protrusion factors⟩* designate the amount that a character should be protruded into the left margin (first value) respectively into the right margin (second value). By default, the values are relative to the character widths, so that a value of 1000 means that the character should be shifted fully into the margin, while, for example, with a value of 50 it would be protruded by 5% of its width. Negative values are admitted, as well as numbers larger than 1000 (but effectively not more than 1em of the font). You can omit either number if the character should not be protruded on that side, but must not drop the separating comma.

Options:

name You may assign a name to the protrusion settings, so that you are able to load it by another list.

load You can load another list (provided, you previously assigned a name to it) before the current list will be loaded, so that the fonts will inherit the values from the loaded list.

Thus, the configuration may be simplified considerably. You can for instance create a default list for a font; settings for other shapes or series can then load these settings, and extend or overwrite them (since the value that comes last will take precedence). Font settings will be loaded recursively. The following options will affect all loaded lists:

factor This option can be used to influence all protrusion factors of the list, overriding any global factor setting (see section 3.2). For instance, if you want fonts in larger sizes to be protruded less, you could load the normal lists, just with a different factor applied to them:

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

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

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

⁵ 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.

inputenc Select an input encoding that should apply to this list, regardless of what the document's input encoding is. You may specify any encoding that can be loaded via the `inputenc` package, e. g., `ansinew`, `koi8-r`, `utf8`, with the exception of `uft8x`.

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

5.2 Font Expansion

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

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

The *expansion settings* consist of $\langle \text{character} \rangle = \langle \text{expansion factor} \rangle$ 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.

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: pdfTeX 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:

```

\SetExpansion
[ stretch = 30,
  shrink = 60,
  step = 5 ]
{ encoding = *,
  size = 10.001 }
{ }
\newcommand{\expandpar}[1]{%6
  \fontsize{10.001}{\baselineskip}\selectfont #1\par}}
% ...
\expandpar{This paragraph contains an 'unnecessary' widow.}

```

factor This option provides a different method to alter expansion settings for certain fonts, working around another restriction of pdf \TeX : 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 Tracking

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

An important typographic technique – which was missing in \TeX for a long time – is tracking, i. e., the uniform addition or subtraction of letter space to/from all the characters in a font. For example, it is good typographic practice to slightly

⁶ 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.

space out pieces of text set in all capitals or small capitals (as in this document), and to increase letter spacing of smaller and decrease that of larger type.⁷ With pdf \TeX 1.40 and `microtype`, this is now possible, and moreover, easy to achieve. For instance, to increase the letter spacing of all fonts smaller than `\small`, and also space out the small-caps font, you could specify something like the following:

```
\usepackage[tracking=true]{microtype}
\DeclareMicrotypeSet*[tracking]{my}
{ encoding = *,
  size      = -small,
  font      = */*/*/sc/* }
\SetTracking{ encoding = *, shape = sc }{ 50 }
\SetTracking{ encoding = *, size = footnotesize }{ 70 }
\SetTracking{ encoding = *, size = -footnotesize }{ 100 }
```

In the above example, the small-caps font would be spaced out by 50/1000em (0.025em on each side of each character), fonts in `\footnotesize` would be spaced out by 0.07em, and even smaller type by 0.1em. The amount of letter spacing is given in thousandth of 1em. The interword space will be scaled accordingly.

Letterspaced fonts for which you do not specify an amount will be spaced out by the default of 0.1em (adjustable with the package option `letterspace`, see section 3.5). Suppose your editor wants you to shorten your 1000 pages chef-d'œuvre by 20 pages, you could load `microtype` with (fingers crossed):

```
\usepackage[tracking=alltext,letterspace=-40]{microtype}
```

Options:

name, unit, context These options serve the same function as in the previous configuration commands. The unit may be any dimension, default is 1em.

5.4 Interword Spacing

`\SetExtraSpacing` [*options*] {*set of fonts*} {*spacing settings*}

This command allows you to fine tune the interword spacing (also known as glue). A preliminary remark on what a ‘space’ is may be in order: Between two words, \TeX will insert a so called glue, which is characterised by three parameters – the normal distance between two words, the maximum amount of space that may be added to it, and the maximum amount that may be subtracted. The latter two parameters come into effect whenever \TeX tries to break a paragraph into lines and does not succeed; it can then stretch or shrink the spaces between words. These three parameters are specific to each font.

On top of these glue dimensions, \TeX has the concept of ‘space factors’. They may be used to increase the space after certain characters, most prominently the punctuation characters. If pdf \TeX ’s additional spacing adjustment is in effect, space factors are ignored, since it may be considered an extension to space factors with much finer control.

⁷ For full-featured fonts like Computer Modern, this is usually not necessary, though, since they come in optical sizes, and the tracking of the small-capitals font is already adjusted.

The *spacing settings* are declared as pairs of $\langle \text{character} \rangle = \langle \text{spacing factors} \rangle$, where the latter consist of three numbers: first, the additional kern inserted after this character if it appears before an interword space, second, the additional stretch amount, and third, the additional shrink amount. All values may also be negative, in which case the dimensions will be decreased. Not all values have to be specified, however, the settings must contain the two separating commas.

Options:

name, **load**, **factor**, **inputenc**, **context** These options serve the same function as in the previous configuration commands.

unit Like in `\SetProtrusion`, you can specify the unit by which the specified numbers are measured. Possible values are: `character`, a $\langle \text{dimension} \rangle$ and, additionally, `space`. The latter will measure the values in thousandths of the respective space dimension set by the font. By default, the unit is measured by the space dimensions. For example, with these settings:

```
\SetExtraSpacing
[ unit = space ] % default
{ font = */*/*/*/* }
{
  . = {1000,1000,1000},
}
```

the space inserted after a full stop would be doubled (technically speaking: $2 \times \text{\fontdimen 2}$), as well as the maximum stretch and the maximum shrink amount of the interword space (`\fontdimen 3` and `4`). As another example, setting all three value to `-1000` would completely cancel a space after the respective character.

preset Preset all characters to the specified three values, possibly scaled by a factor, and measured by a given unit (except `character`).

5.5 Additional Kerning

`\SetExtraKerning` [$\langle \text{options} \rangle$] [$\langle \text{set of fonts} \rangle$] [$\langle \text{kerning settings} \rangle$]

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

I should not neglect to mention a limitation of this additional kerning: if a word *immediately follows* such a kern (i. e., not separated by a space), hyphenation will be inhibited, unless you insert the breakpoints with `\-` manually. This restriction of pdf_T_E_X will hopefully be lifted one day.

The *kerning settings* are specified as pairs of $\langle \text{character} \rangle = \langle \text{kerning values} \rangle$, where the latter consist of two values: the kerning added before the character, and the kerning appended after the respective character. Once again, either value may be omitted, but not the separating comma.

Options:

name, **load**, **factor**, **preset**, **inputenc** These options serve the same function as in the previous configuration commands.

unit Admissible values are: space, character and a *<dimension>*. By default, the values are relative to 1em.

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

5.6 Character Inheritance

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

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

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

5.7 Configuration Files

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

If you are embarking on creating new expansion and protrusion settings for a font family, you should put them into a separate file, whose name must be: ‘`mt-.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 Minion⁸, for Bitstream Charter, the AMS symbols and Euler fonts and Euro symbol fonts (Adobe, ITC and marvosym). Table 2 lists them all.

⁸ By courtesy of Harald Harders (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) ^a	n, (it, sl, sc) ^a
Computer Modern Roman (cmr) ^b	OT1, OT4, T1, T5, LY1, TS1	n, it, sl, sc
Bitstream Charter (bch) ^c	OT1, T1, T5, LY1, TS1	n, it, (sl) ^d , sc
Adobe Garamond (pad, padx, padj)	OT1, T1, LY1, TS1	n, it, (sl) ^d , sc
URW Garamond (ugm) ^e	OT1, T1, TS1	n, it
Adobe Minion (pmnx, pmnj)	OT1, T1, LY1, TS1	n, it, (sl) ^d , sc, si
Palatino (ppl, pplx, pplj) ^f	OT1, OT4, T1, LY1, (TS1) ^a	n, it, (sl) ^d , sc
Times (ptm, ptmx, ptmj) ^g	OT1, OT4, T1, LY1, QX, (TS1) ^a	n, it, (sl) ^d , sc
Computer Modern math (cmsy, cmm)	OML/OMS	n/it
AMS symbols (msa, msb)	U	n
Euler (eur, eus, euf) ^h	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 (qp1)

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

^h Alias: eulervm (zeur, zeus)

`\DeclareMicrotypeAlias` {} {<alias font>}

You may use this command for fonts that are very similar, or actually the same (for instance if you did not stick to the Berry naming scheme when installing the font). An example would be the Latin Modern fonts which are clones of the Computer Modern fonts, so that it is not necessary to create new settings for them – you could say:

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

which would make the package, whenever it encounters the font `lmr` and does not find settings for it, also try the font `cmr`. In fact, you will find this very line, along with some others, in the default configuration file.

`\LoadMicrotypeFile` {}

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.⁹ This command will load the file `mt-.cfg`.

⁹ Font package authors might also want to have a look at the hook `\Microtype@Hook`, described in the implementation part, section 14.4.3.

6 Context-sensitive Setup

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

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

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

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

```
\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^AT_EX 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.

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

```
\microtypecontext{kerning=french}
```

to the preamble. This would have the effect that kerning settings for the French context would be applied to the document. Should parts of the document be in English, you could insert

```
\microtypecontext{kerning=}
```

to reset the context, so that the punctuation characters in these parts will not receive any extra kerning.

Instead of adding these commands manually to your document, you may also load `microtype` with the `babel` option. The current language will then be automatically detected and the contexts set accordingly.

```
\DeclareMicrotypeBabelHook {⟨list of babel languages⟩} {⟨context list⟩}
```

Naturally, `microtype` does not know about the typographic specialties of every language. This command is a means of teaching it how to adjust the context when a particular language is selected. The main configuration file contains among others the following declaration:

```
\DeclareMicrotypeBabelHook
{french,français,acadian,canadien}
{kerning=french, spacing=}
```

Consequently, whenever you switch to the French language, the kerning context will be changed to ‘french’ and the spacing context will be reset. This hook only has an effect if the package has been loaded with the `babel` option (see section 3.5). Currently, `microtype` supports French and Turkish kerning and English spacing (aka. `\nonfrenchspacing`). For unknown languages, all contexts will be reset.

7 Letterspacing revisited

```
\textls [⟨amount⟩] {⟨general text⟩}
```

`\textls*` While the tracking feature, described in section 5.3, will apply to sets of fonts, you may also want to letterspace shorter pieces of text, regardless of the font in which they are typeset.¹⁰ For such ad-hoc letterspacing, `microtype` introduces two commands that can be used in the same way as L^AT_EX’s text commands: `\textls` expects the text in the mandatory argument, while `\lsstyle` will switch on letterspacing for all subsequent fonts until the end of the current group. The starred version of `\textls` does not add any extra space before or after the text, which may be useful, e. g., for section titles. By default, each character will be spaced out by 100/1000em (0.05em on each side); this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option (see section 3.5).

```
\lslig {⟨ligature⟩}
```

One side-effect of letterspacing is that ligatures will be disabled. While this is usually typographically correct, it is not in the case of Fraktur fonts, where the ligatures ‘ch’, ‘ck’, ‘tz’ and ‘sz’ (ß) should not be broken up. You can protect ligatures (including ‘s:’, ‘a’, ‘*a’ etc.) with the `\lslig` command:

```
\textfrac{\lsstyle S\lslig{ch}\mu\lslig{tz}\fle\lslig{ck}} % Schmuckfleck
```

¹⁰ Letterspacing should be used cautiously. Renowned typographers have compared letterspacing lower-case text to stealing sheep (a capital offence in Britain in the 19th century). Unless you know what you are doing, you should probably only letterspace small-capitals or all-capitals. Another just cause may be emphasis in texts typeset in Fraktur fonts.

`letterspace.sty` These three commands (plus the `letterspace` option) are also available with the `letterspace` package, which is in fact a much stripped-down version of `microtype`, omitting support for all the other extensions. If you find that you don't need `microtype`'s specialties and it is only slowing down the compilation, you should use the `letterspace` package instead (which won't work together with `microtype`; it does not contain anything that `microtype` doesn't).

8 Disabling Ligatures

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

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

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

This command may only be used in the preamble.

9 Hints and Caveats

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

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

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

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

electronically. With pdf \TeX 1.40, which uses a different technique of expansion, the file size increase can be neglected.

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.

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

You might want to disable protrusion in 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:¹¹

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

Compatibility. The package should work happily together with all other \LaTeX packages (except `pdfcpot`). However, life isn't perfect, so problems are to be expected. Currently, I am only aware of the following issue:

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

¹¹ If you are using the `fancyvrb` or the `listings` package, this is not necessary, since their implementation of the corresponding environments will inhibit protrusion anyway.

Possible error messages and how to get rid of them:

- ! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.
This error message will occur if you are trying to employ font expansion while creating DVI output. Remember, that *automatic* font expansion only works when running pdf \TeX in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your \TeX system.
- Warning: pdf \LaTeX : font ptmr8r cannot be expanded (not an included Type1 font)
and the PDF viewer complains about a missing font, e. g., Adobe Reader thusly:
Could not find a font in the Resources dictionary - using Helvetica instead.
With pdf \TeX versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your \TeX system is not set up to embed (or ‘download’) the base PostScript fonts (e. g., Times, Helvetica, Courier). In most \TeX distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to true.
- Warning: pdf \LaTeX (file ecrm1000+20): Font ecrm1000+20 at 1200 not found
Furthermore, pdf \TeX older than 1.40 requires Type 1 fonts for automatic font expansion. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. With older pdf \TeX versions, this is only possible if you manually create expanded instances of the fonts.
- ! Font T1/cmr/m/n/10=ecrm1000 at 10.0pt not loaded: Not enough room left.
Memory parameter ‘font_mem_size’ too small.
- ! TeX capacity exceeded, sorry [maximum internal font number (font_max)=2000].
Memory parameter ‘font_max’ too small.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf_mem_size)=65536].
Memory parameter ‘pdf_mem_size’ too small (pdf \TeX versions older than 1.30).
When applying micro-typographic enhancement to a large document with a lot of fonts, pdf \TeX may be running out of some kind of memory. It may be increased by setting the respective parameter to a larger value. For web2c-based systems, change the settings in `texmf.cnf`, for MiK \TeX , in the file `miktex.ini` (2.4) resp. `pdf \LaTeX .ini` (2.5).
- pdf \TeX warning (font expansion): font should be expanded before its first use
This warning will occur if tracking *and* expansion is applied to a font. It is harmless and can be ignored.

10 Contributions

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

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

11 Acknowledgments

This package would be pointless if *Hàn Thế Thành* hadn't created the pdfT_EX programme in the first place, which introduced the micro-typographic extensions and made them available to the T_EX world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004]. I also thank him and the rest of the pdfT_EX team for refuting the idea that T_EX is dead, and for fixing the bugs I find.

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

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

Additionally, the following people have reported bugs, made suggestions or helped otherwise (in chronological order): *Tom Kink*, *Herb Schulz*, *Michael Hoppe*, *Gary L. Gray*, *Georg Verweyen*, *Christoph Bier*, *Peter Muthesius*, *Bernard Gaulle*, *Adam Kucharczyk*, *Mark Rossi*, *Stephan Hennig*, *Michael Zedler*, *Herbert Voß*, *Ralf Stubner*, *Holger Uhr*, *Peter Dyballa*, *Steven Bath*, *Daniel Flipo*, *Michalis Miatidis* and *Sven Naumann*.

12 References

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

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

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, Martin Schröder, *The pdfT_EX user manual*, 1 January 2007. (Available from CTAN at `/systems/pdftex/`; latest version at <http://sarovar.org/projects/pdftex/>)

L^AT_EX3 Project Team, *L^AT_EX 2_ε font selection*, 27 November 2005. (Available from CTAN at `/macros/latex/doc/fntguide.pdf`)

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

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

13 Short History

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

2.1 (2007/01/21)

New command `\slig` to protect ligatures in letterspaced text (see section 7)

2.0 (2007/01/14)

Support for the new extensions of pdfTeX version 1.40: tracking/letterspacing, adjustment of interword spacing (`glue`), and additional kerning (new commands `\SetTracking`, `\SetExtraSpacing`, `\SetExtraKerning`; new options ‘tracking’, ‘spacing’, ‘kerning’; see sections 5.3, 5.4 and 5.5)

New commands `\textls` and `\lststyle` for letterspacing, new option ‘letterspace’ (see sections 3.4 and 7)

New option ‘babel’ for automatic micro-typographic adjustment to the selected language (see sections 3.5 and 6)

New font sets: ‘smallcaps’, ‘footnotesize’, ‘scriptsize’ (see section 4 and table 1)

New package ‘letterspace’ providing the commands for robust and hyphenatable letterspacing

1.9e (2006/07/28)

New key ‘inputenc’ to specify the lists’ input encodings (see section 5)

Protrusion settings for Euler math fonts

1.9d (2006/05/05)

Support for the Central European QX encoding (inheritance, generic protrusion settings, contributed by Maciej Eder; protrusion settings for Times)

Protrusion settings for various Euro symbol fonts (Adobe, ITC, marvosym)

Support for Unicode input in the configuration (`inputenc/utf8`)

1.9c (2006/02/02)

Protrusion settings for URW Garamond

1.9a (2005/12/05)

Defer setup until the end of the preamble; consequently, no need to change font defaults before loading microtype, or to put it the other way round, microtype may now be loaded at any time

Inside the preamble, `\microtypesetup` accepts all package options (see section 3.6)

Protrusion settings for T5 encoded Charter

1.9 (2005/10/28)

New command `\DisableLigatures` to disable ligatures of fonts (requires pdfTeX version 1.30 or later; see section 8)

New command `\microtypecontext` to change the configuration context; new key ‘context’ for the configuration commands (see section 6)

New key ‘font’ to add single fonts to the font sets (see section 4)

New key ‘preset’ to set all characters to the specified value before loading the lists

Value ‘relative’ renamed to ‘character’ for ‘unit’ keys
 Support for the Polish OT4 encoding (protrusion, expansion, inheritance)
 Support for the Vietnamese T5 encoding (protrusion, expansion, inheritance)
 ‘DVIoutput’ option will work with T_EXLive 2004

1.8 (2005/06/23)

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

1.7 (2005/03/23)

Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings (see sections 4 and 5)
 Always take font size into account when trying to find protrusion resp. expansion settings for a given font (see section 5)
 New command `\LoadMicrotypeFile` to load a font configuration file manually (see section 5.7)
 Hook `\Microtype@Hook` for font package authors (see section 14.4.3)
 New option ‘verbose=errors’ to turn all warnings into errors
 Disable expansion inside `\showhyphens`
 Warning when running in draft mode

1.6 (2005/01/24)

New option ‘factor’ to influence protrusion resp. expansion of all characters of a font or font set (see sections 3.2 and 5)
 When pdfT_EX 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-T_EX extensions, if available

1.5 (2004/12/15)

When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled (see section 3.1)
 New option ‘selected’ to enable selected expansion (see sections 3.3 and 5.2); default is: false

New default for expansion option ‘step’: 4 ($\min(\text{stretch}, \text{shrink})/5$) (see section 3.3)

Protrusion settings for Bitstream Charter

1.4b (2004/11/26)

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

1.4 (2004/11/12)

Set up fonts independently from \LaTeX font loading (therefore, no risk of overlooking fonts anymore, and the package may be loaded at any time)

`\microtypesetup` now sets the correct level of protrusion (see chapter 3.6)

New option: ‘final’

1.2 (2004/10/03)

New font sets: ‘allmath’ and ‘basicmath’ (see section 4 and table 1)

Protrusion settings for Computer Modern Roman math symbols

Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond

If an alias font name is specified, it will be used as an alternative, not as a replacement (see section 5.7)

1.1 (2004/09/21)

Protrusion settings for Adobe Minion, contributed by Harald Harders

New command: `\DeclareCharacterInheritance` (see section 5.6)

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

Configuration file names in lowercase (see section 5.7)

1.0 (2004/09/11)

First CTAN release

14 Implementation

The docstrip modules in this file are:

driver: The documentation driver, only visible in the dtx file.

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

debug: Code for additional output in the log file.

Used for – surprise! – debugging purposes.

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

config: Surrounds all configuration modules.

cfg-t: Surrounds (Latin) text configurations.

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

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

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

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

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

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

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

Contributed by *Harald Harders*.

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

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

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

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

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

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

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

cfg-e: Surrounds Euro symbol configurations.

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

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

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

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

And now for something completely different.

```
1 <{*package|letterspace}>
2 <{*package}>
```

14.1 Preliminaries

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

```
3 \newcommand*\DeclareMicrotypeSet[3] [] {}
```

```

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

```

This command also has a starred version.

```
26 \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.

```

27 \@onlypreamble\DeclareMicrotypeSet
28 \@onlypreamble\UseMicrotypeSet
29 \@onlypreamble\DeclareMicrotypeSetDefault
30 \@onlypreamble\DisableLigatures
31 \@onlypreamble\DeclareMicrotypeBabelHook

```

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

```

32 \def\MT@old@cmd#1#2{%
33   \newcommand*#1{\MT@warning{%
34     \string#1 is deprecated. Please use\MessageBreak
35     \string#2 instead}%
36   \let #1#2#2}}
37 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
38 \MT@old@cmd\DeclareMicroTypeSet \DeclareMicrotypeSet
39 \MT@old@cmd\UseMicroTypeSet \UseMicrotypeSet
40 \MT@old@cmd\LoadMicroTypeFile \LoadMicrotypeFile

```

\MT@MT This is us.

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

\MT@info Communicate.

```

\MT@info@nl 42 <*/debug>
\MT@vinfo 43 \def\MT@info{\PackageInfo\MT@MT}
\MT@warning 44 \def\MT@info@nl#1{\MT@info{#1@gobble}}
\MT@warning@nl 45 \let\MT@vinfo@gobble
\MT@warn@err 46 \def\MT@warning{\PackageWarning\MT@MT}
\MT@error 47 \def\MT@warning@nl#1{\MT@warning{#1@gobble}}
48 </!debug>
49 </package>
50 <*letterspace>
51 \def\MT@warning{\PackageWarning{letterspace}}

```

```

52 \def\MT@warning@n1#1{\MT@warning{#1\@gobble}}
53 /letterspace
54 *package
55 \def\MT@warn@err#1{\MT@error{#1}}%
56 This error message appears because you loaded the \MT@MT\MessageBreak
57 package with the option `verbose=errors'. Consult the documentation\MessageBreak
58 in \MT@MT.pdf to find out what went wrong.}}
59 \def\MT@error{\PackageError\MT@MT}

```

14.1.1 Debugging

\tracingmicrotype Cases for \tracingmicrotype:

\MT@dinfo 0: almost none
\MT@dinfo@n1 1: + sets & lists
2: + heirs
3: + slots
4: + factors

```

60 *debug
61 \newcount\tracingmicrotype
62 \tracingmicrotype=\tw@
63 \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}%
64 \def\MT@info@n1#1{\PackageInfo\MT@MT{#1\@gobble}\MT@addto@annot{#1}}%
65 \let\MT@vinfo\MT@info@n1
66 \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}%
67 \def\MT@warning@n1#1{\PackageWarning\MT@MT{#1\@gobble}\MT@addto@annot{Warning: #1}}%
68 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
69 \def\MT@dinfo@n1#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@n1{#2}\fi}

```

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

Cases for \tracingmicrotypeinpdf:

1: show new fonts
2: + show known fonts
70 \newcount\tracingmicrotypeinpdf

Let's see how it works ...

```
\tracingmicrotypeinpdf=2
```

\MT@pdf@annot During font setup, we save the text for the popup in \MT@pdf@annot. (This requires pdfTeX ≥ 1.30.)

```

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

```

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

```
76 \newif\iftracingmicrotypeinpdfall
```

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

```

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

```

14.1.2 Requirements

`\MT@pdf@tex@no` pdf_T_EX'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_T_EX we're using, if any. `\MT@pdf@tex@no` will be used throughout the package to respectively do the right thing.

Currently, we have to distinguish between six cases for pdf_T_EX:

- 0: not running pdf_T_EX
- 1: pdf_T_EX (< 0.14f)
- 2: + micro-typographic extensions (0.14f, 0.14g)
- 3: + protrusion relative to 1em (\geq 0.14h)
- 4: + automatic font expansion; default `\efcode = 1000` (\geq 1.20)
- 5: + `\(left,right)marginkern`; `\pdfnoligatures`; `\pdfstrcmp`; `\pdfescapestring` (\geq 1.30)
- 6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`¹² (\geq 1.40)

```

93 </package>
94 \def\MT@pdf@tex@no{0}

```

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

```

95 \ifx\normalpdf@texversion\undefined \else
96   \let\pdf@texversion\normalpdf@texversion
97   \let\pdf@texrevision\normalpdf@texrevision
98   \let\pdf@output\normalpdf@output
99 \fi

```

Old packages might have let `\pdf@texversion` to `\relax`.

¹² This command was actually introduced in 1.30, but was still buggy then.


```

100 \ifx\pdfTeXversion\undefined \else
101   \ifx\pdfTeXversion\relax \else
102   <debug> \MT@info@nl{0}{running pdfTeX \the\pdfTeXversion(\pdfTeXrevision)}
103   \def\MT@pdfTeX@no{6}
104   \ifnum\pdfTeXversion < 140
105     \def\MT@pdfTeX@no{5}
106   <package>
107     \ifnum\pdfTeXversion < 130
108       \def\MT@pdfTeX@no{4}
109     \ifnum\pdfTeXversion < 120
110       \def\MT@pdfTeX@no{3}
111     \ifnum\pdfTeXversion = 14
112       \ifnum \expandafter`\pdfTeXrevision < `h
113         \def\MT@pdfTeX@no{2}
114       \ifnum \expandafter`\pdfTeXrevision < `f
115         \def\MT@pdfTeX@no{1}
116     \fi
117   \fi
118   \else
119     \ifnum\pdfTeXversion < 14
120       \def\MT@pdfTeX@no{1}
121     \fi
122   \fi
123 \fi
124 \fi
125 </package>
126 \fi
127 \fi
128 \fi
129 <debug>\MT@info@nl{0}{pdfTeX no.: \number\MT@pdfTeX@no}

```

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

```

130 <letterspace>\ifnum\MT@pdfTeX@no<6
131 <package>\ifnum\MT@pdfTeX@no<2
132   \AtEndOfPackage{\let\unprocessedoptions\relax}
133   \let\CurrentOption\empty
134   \MT@warning@nl{%
135     \ifcase\MT@pdfTeX@no
136       You don't seem to be using pdfTeX.\MessageBreak
137     \else
138   <package>
139     You are using a pdfTeX version older than 0.14f.\MessageBreak
140     ~\MT@MT' won't work with such antiquated versions.\MessageBreak
141     Please install a newer version of pdfTeX.\MessageBreak
142   </package>
143   <*letterspace>
144     You are using a pdfTeX version older than 1.40.\MessageBreak
145     Sorry, but robust letterspacing doesn't work with\MessageBreak
146     this version. Please upgrade%
147   </letterspace>
148   \fi
149 <package> All micro-typographic features will be disabled%
150 }
151 \endinput\fi
152 <package>

```

Still there? Then we can begin:

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

```

153 \def\MT@catcodes{%
154   \catcode`\^7 %
155   \@makeother\-%
156   \@makeother\=%
157   \@makeother\*%
158   \@makeother\+%
159   \@makeother\,%
160   \@makeother\/%
161   \@makeother\`%
162   \@makeother\'%
163   \@makeother\"%
164   \@makeother\!%
165 }

```

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

```

166 \def\MT@restore@catcodes#1{%
167   \ifx\relax#1\else
168     \noexpand\catcode`\noexpand#1\the\catcode`#1\relax
169     \expandafter\MT@restore@catcodes
170   \fi
171 }
172 \edef\MT@restore@catcodes{\MT@restore@catcodes\^-\= \* \+ \, \/ \` \' \'! \relax}
173 \MT@catcodes
174 \AtEndOfPackage{\MT@restore@catcodes}

```

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

```

175 </package>
176 \RequirePackage{keyval}[1997/11/10]
177 <*package>

```

\MT@toks We need a token register.

```
178 \newtoks\MT@toks
```

\ifMT@if@ A scratch if.

```
179 \newif\ifMT@if@
```

14.1.3 Declarations

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

```

\ifMT@expansion 180 \newif\ifMT@protrusion
\ifMT@auto       181 \newif\ifMT@expansion
\ifMT@selected   182 \newif\ifMT@auto
\ifMT@noligatures 183 \newif\ifMT@selected
\ifMT@draft      184 \newif\ifMT@noligatures
\ifMT@spacing    185 \newif\ifMT@draft
\ifMT@kerning    186 \newif\ifMT@spacing
\ifMT@tracking   187 \newif\ifMT@kerning
\ifMT@tracking   188 \newif\ifMT@tracking
\ifMT@tracking   189 \newif\ifMT@babel

```

\MT@pr@babel ... and numbers.

```

\MT@ex@level 190 \let\MT@pr@level\tw@
\MT@pr@factor 191 \let\MT@ex@level\tw@
\MT@ex@factor 192 \let\MT@pr@factor\@m
\MT@sp@factor 193 \let\MT@ex@factor\@m
\MT@kn@factor 194 \let\MT@sp@factor\@m
\MT@kn@factor 195 \let\MT@kn@factor\@m

```

\MT@pr@unit Default unit for protrusion settings is character width, for spacing space, for kerning (and tracking) 1em.

```

\MT@sp@unit
\MT@kn@unit
\MT@stretch
\MT@shrink
\MT@step

```

```

196 \let\MT@pr@unit\@empty
197 \let\MT@sp@unit\m@ne
198 \def\MT@kn@unit{1em}

```

Expansion settings.

```

199 \let\MT@stretch\m@ne
200 \let\MT@shrink \m@ne
201 \let\MT@step \m@ne

```

\MT@pr@min Minimum and maximum values allowed by pdf_{TEX}.

```

\MT@pr@max 202 \def\MT@pr@min{-\@m}
\MT@ex@min 203 \let\MT@pr@max\@m
\MT@ex@max 204 \let\MT@ex@min\z@
\MT@sp@min 205 \let\MT@ex@max\@m
\MT@sp@max 206 \def\MT@sp@min{-\@m}
\MT@kn@min 207 \let\MT@sp@max\@m
\MT@kn@max 208 \def\MT@kn@min{-\@m}
\MT@tr@min 209 \let\MT@kn@max\@m
\MT@tr@max 210 </package>
211 \def\MT@tr@min{-\@m}
212 \let\MT@tr@max\@m
213 <*package>

```

\MT@factor@default Default factor.

```

214 \def\MT@factor@default{1000 }

```

\MT@stretch@default Default values for expansion.

```

\MT@shrink@default 215 \def\MT@stretch@default{20 }
\MT@step@default 216 \def\MT@shrink@default{20 }
217 \def\MT@step@default{4 }

```

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

```

\MT@letterspace@default 218 </package>
219 \let\MT@letterspace\m@ne
220 \def\MT@letterspace@default{100}
221 <*package>

```

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

```

222 \newif\ifMT@document

```

14.1.4 Auxiliary Macros

\MT@requires@etex For definitions that depend on e-_{TEX} features.

```

223 \expandafter\let\expandafter\MT@requires@etex
224 \ifcase 0%
225 \ifx\TeXversion\@undefined 1\else
226 \ifx\TeXversion\relax 1\else
227 \ifcase\TeXversion 1\fi
228 \fi
229 \fi\space
230 \@firstoftwo
231 \else
232 \@secondoftwo
233 \fi
234 <debug>\MT@dinfo@n1{0}{\MT@requires@etex}{\not }running etex}

```

\MT@requires@pdftex For definitions that depend on a particular pdf_{TEX} version.

```

235 \def\MT@requires@pdftex#1{%
236 \ifnum\MT@pdftex@no<#1
237 \expandafter\@secondoftwo

```

```

238 \else
239 \expandafter\@firstoftwo
240 \fi
241 }

\MT@gllet The forgotten primitive.
242 </package>
243 \def\MT@gllet{\global\let}
244 <*package>

\MT@def@n This is \@namedef and global.
\MT@gdef@n 245 \def\MT@def@n#1{\expandafter\def\csname #1\endcsname}
246 \def\MT@gdef@n#1{\expandafter\gdef\csname #1\endcsname}

\MT@edef@n Its expanding versions.
\MT@xdef@n 247 </package>
248 \def\MT@edef@n#1{\expandafter\edef\csname #1\endcsname}
249 <*package>
250 \def\MT@xdef@n#1{\expandafter\xdef\csname #1\endcsname}

\MT@let@nc \let a \csname sequence to a command.
251 </package>
252 \def\MT@let@nc#1{\expandafter\let\csname #1\endcsname}
253 <*package>

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

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

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

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

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

\MT@exp@two@n Expand the next two tokens after <#1> once and enclose them in braces.
261 \def\MT@exp@two@n#1#2#3{%
262 \expandafter\expandafter\expandafter
263 #1\expandafter\expandafter\expandafter
264 {\expandafter#2\expandafter}\expandafter{#3}}

You do not wonder why \MT@exp@one@c doesn't exist, do you?
\MT@ifdefined@c@T Wrapper for testing whether command resp. \csname sequence is defined. If we
\MT@ifdefined@c@TF are running e-TeX, we will use its primitives \ifdefined and \ifcsname, which
\MT@ifdefined@n@T decreases memory use substantially.
\MT@ifdefined@n@TF 265 \MT@requires@etex{
266 \def\MT@ifdefined@c@T#1{\ifdefined#1%
267 \expandafter\@firstofone\else\expandafter\@gobble\fi
268 }
269 </package>
270 \def\MT@ifdefined@c@TF#1{\ifdefined#1%
271 \expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
272 }

```

```

273 \def\MT@ifdefined@n@T#1{\ifcsname#1\endcsname
274 \expandafter\@firstofone\else\expandafter\@gobble\fi
275 }
276 <package>
277 \def\MT@ifdefined@n@TF#1{\ifcsname#1\endcsname
278 \expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
279 }
280 }{
281 \def\MT@ifdefined@c@T#1{\ifx#1\@undefined
282 \expandafter\@gobble\else\expandafter\@firstofone\fi
283 }
284 \def\MT@ifdefined@c@TF#1{\ifx#1\@undefined
285 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
286 }
287 \def\MT@ifdefined@n@T#1{\begingroup\MT@exp@two@c\endgroup
288 \ifx\csname #1\endcsname\relax
289 \expandafter\@gobble\else\expandafter\@firstofone\fi
290 }
291 \def\MT@ifdefined@n@TF#1{\begingroup\MT@exp@two@c\endgroup
292 \ifx\csname #1\endcsname\relax
293 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
294 }
295 }

```

`\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`

```

296 \MT@requires@etex{
297 \def\MT@detokenize@n#1{\detokenize\expandafter{\string#1}}
298 \def\MT@detokenize@c#1{\detokenize
299 \expandafter\expandafter\expandafter{\expandafter\string#1}}
300 }{
301 \def\MT@detokenize@n#1{\string#1}
302 \def\MT@detokenize@c#1{\MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty}
303 }

```

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

```

304 </package>
305 \begingroup
306 \catcode`\%=12
307 \catcode`\&=14
308 \gdef\MT@ifempty#1{&
309 \if %#1%&
310 \expandafter\@firstoftwo
311 \else
312 \expandafter\@secondoftwo
313 \fi
314 }
315 \endgroup

```

`\MT@ifint` Test whether argument is an integer, using an old trick by Mr. Arseneau, or the latest and greatest from pdf \TeX (which also allows negative numbers, as required by the `letterspace` option).

```

316 <package>\MT@requires@pdfTeX6{
317 \def\MT@ifint#1{%
318 \ifcase\pdfmatch{^-[0-9]+ *$}\{#1}\relax
319 \expandafter\@secondoftwo
320 \else
321 \expandafter\@firstoftwo
322 \fi
323 }

```

```

324 <package>
325 {}{
326   \def\MT@ifint#1{%
327     \if!\ifnum9<1#1!\else?\fi
328     \expandafter\@firstoftwo
329     \else
330     \expandafter\@secondoftwo
331     \fi
332   }
333 }

```

\MT@ifdimen Test whether argument is dimension (or number). (nd and nc are new Didot resp. Cicero, added in pdfTeX 1.30; px is a pixel, it seems.)

```

334 \MT@requires@pdftex6{
335   \def\MT@ifdimen#1{%
336     \ifcase\pdfmatch{^[0-9]+([.],[0-9]+)?|[.],[0-9]+)%
337                     (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
338     \expandafter\@secondoftwo
339     \else
340     \expandafter\@firstoftwo
341     \fi
342   }
343 }{
344   \def\MT@ifdimen#1{%
345     \setbox\z@=\hbox{%
346       \MT@count=1#1\relax
347       \ifnum\MT@count=\@ne
348         \aftergroup\@secondoftwo
349       \else
350         \aftergroup\@firstoftwo
351       \fi
352     }%
353   }
354 }

```

\MT@ifdim Test floating point numbers.

```

355 \def\MT@ifdim#1#2#3{%
356   \ifdim #1\p@ #2 #3\p@
357     \expandafter\@firstoftwo
358   \else
359     \expandafter\@secondoftwo
360   \fi
361 }

```

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

```

362 \MT@requires@pdftex5{
363   \def\MT@ifstreq#1#2{%
364     \ifcase\pdfstrcmp{#1}{#2}\relax
365     \expandafter\@firstoftwo
366     \else
367     \expandafter\@secondoftwo
368     \fi
369   }
370 }{
371   \def\MT@ifstreq#1#2{%
372     \edef\x{#1}%
373     \edef\y{#2}%
374     \ifx\x\y
375       \expandafter\@firstoftwo
376     \else
377       \expandafter\@secondoftwo
378     \fi

```

```

379   }
380 }

\MT@xadd Add item to a list.
381 \def\MT@xadd#1#2{%
382   \ifx#1\relax
383     \xdef#1{#2}%
384   \else
385     \xdef#1{#1#2}%
386   \fi
387 }

\MT@xaddb Add item to the beginning.
388 \def\MT@xaddb#1#2{%
389   \ifx#1\relax
390     \xdef#1{#2}%
391   \else
392     \xdef#1{#2#1}%
393   \fi
394 }

\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@ 395 </package>
\MT@clist@function 396 \def\MT@map@clist@n#1#2{%
\MT@clist@break 397   \ifx\@empty#1\else
398     \def\MT@clist@function#1{#2}%
399     \MT@map@clist@#1,\@nil,\@nnil
400   \fi
401 }
402 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
403 \def\MT@map@clist@#1,{%
404   \ifx\@nil#1%
405     \expandafter\MT@clist@break
406   \fi
407   \MT@clist@function{#1}%
408   \MT@map@clist@
409 }
410 \let\MT@clist@function\@gobble
411 \def\MT@clist@break#1\@nnil{}
412 <*/package>

\MT@map@tlist@n Execute <#2> on all elements of the token list <#1>. \MT@tlist@break can be used
\MT@map@tlist@c to jump out of the loop.
\MT@map@tlist@ 413 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break 414 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
415 \def\MT@map@tlist@#1#2{%
416   \ifx\@nnil#2\else
417     #1{#2}%
418     \expandafter\MT@map@tlist@
419     \expandafter#1%
420   \fi
421 }
422 \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 423 \newif\ifMT@inlist@
424 \def\MT@in@clist#1#2{%
425   \def\x##1,#1,##2##3\@nnil{%
426     \ifx##2\@empty
427       \MT@inlist@false
428     \else

```

```

429     \MT@inlist@true
430     \fi
431 }%
432 \expandafter\x\expandafter,#2,#1,\@empty\@nnil
433 }

\MT@rem@from@clist Remove item <#1> from comma list <#2>. This is basically \@removeelement from
ltxcncrl.dtx. Using \pdfmatch and \pdflastmatch here would be really slow!
434 \def\MT@rem@from@clist#1#2{%
435   \def\x##1,#1,##2\x{##1,##2\y}%
436   \def\y##1,\y##2\y{\ifx,##1\@empty\else##1\fi}%
437   \xdef#2{\MT@exp@two@c\y\x\expandafter,#2,\y,#1,\x}%
438 }

\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.
439 \def\MT@in@tlist#1#2{%
440   \MT@inlist@false
441   \def\x{#1}%
442   \MT@map@tlist@c#2\MT@in@tlist@
443 }
444 \def\MT@in@tlist@#1{%
445   \edef\y{#1}%
446   \ifx\x\y
447     \MT@inlist@true
448     \expandafter\MT@tlist@break
449   \fi
450 }

\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@@ 451 \def\MT@in@rlist#1{%
\MT@size@name 452   \MT@inlist@false
453   \MT@map@tlist@c#1\MT@in@rlist@
454 }
455 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
456 \def\MT@in@rlist@@#1#2#3{%
457   \MT@ifdim{#2}=\m@ne{%
458     \MT@ifdim{#1}=\MT@size
459     \MT@inlist@true
460     \relax
461   }{%
462     \MT@ifdim\MT@size<{#1}\relax{%
463       \MT@ifdim\MT@size<{#2}%
464       \MT@inlist@true
465       \relax
466     }%
467   }%
468   \ifMT@inlist@
469     \def\MT@size@name{#3}%
470     \expandafter\MT@tlist@break
471   \fi
472 }

\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 473 \def\MT@loop#1\MT@repeat{%
474   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
475   \MT@iterate \let\MT@iterate\relax
476 }

```



```

477 \let\MT@repeat\fi
\MT@while@num Execute <#3> from <#1> up to (excluding) <#2>.
478 \def\MT@while@num#1#2#3{%
479   \@tempcnta#1\relax
480   \MT@loop #3%
481   \advance\@tempcnta \@ne
482   \ifnum\@tempcnta < #2\MT@repeat
483 }

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

\MT@count Increment macro <#1> by one. Saves using up too many counters. The e-TeX way is
\MT@increment slightly faster.
485 \newcount\MT@count
486 \MT@requires@etex{
487   \def\MT@increment#1{\edef#1{\number\numexpr #1 + 1\relax}}
488 }{
489   \def\MT@increment#1{%
490     \MT@count=#1\relax
491     \advance\MT@count \@ne
492     \edef#1{\number\MT@count}%
493   }
494 }

\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.
495 \MT@requires@etex{
496   \def\MT@scale#1#2#3{%
497     \ifnum #3 = \z@
498       #1=\numexpr #1 * #2\relax
499     \else
500       #1=\numexpr #1 * #2 / #3\relax
501     \fi
502   }
503 }{
504   \def\MT@scale#1#2#3{%
505     \multiply #1 #2\relax
506     \ifnum #3 = \z@ \else
507       \divide #1 #3\relax
508     \fi
509   }
510 }

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

\MT@abbr@pr Some abbreviations. Thus, we can have short command names but full-length log
\MT@abbr@ex output.
\MT@abbr@pr@c 512 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 513 \def\MT@abbr@ex{expansion}
\MT@abbr@pr@c 514 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@pr@inh 515 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh 516 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@ex@inh 517 \def\MT@abbr@ex@inh{expansion inheritance}
\MT@abbr@nl 518 \def\MT@abbr@nl{noligatures}
\MT@abbr@sp 519 \def\MT@abbr@sp{spacing}
\MT@abbr@sp@c
\MT@abbr@sp@inh
\MT@abbr@kn
\MT@abbr@kn@c
\MT@abbr@kn@inh
\MT@abbr@tr
\MT@abbr@tr@c

```

```

520 \def\MT@abbr@sp@c{interword spacing codes}
521 \def\MT@abbr@sp@inh{interword spacing inheritance}
522 \def\MT@abbr@kn{ Kerning}
523 \def\MT@abbr@kn@c{ Kerning codes}
524 \def\MT@abbr@kn@inh{ Kerning inheritance}
525 \def\MT@abbr@tr{tracking}
526 \def\MT@abbr@tr@c{tracking amount}

\MT@rbba@protrusion These we also need the other way round.
\MT@rbba@expansion 527 \def\MT@rbba@protrusion{pr}
\MT@rbba@spacing 528 \def\MT@rbba@expansion{ex}
\MT@rbba@kerning 529 \def\MT@rbba@spacing{sp}
\MT@rbba@tracking 530 \def\MT@rbba@kerning{kn}
531 \def\MT@rbba@tracking{tr}

\MT@features We can work on these lists to save some guards in the dtx file.
\MT@features@long 532 \def\MT@features{pr,ex,sp,kn,tr}
533 \def\MT@features@long{protrusion,expansion,spacing,kerning,tracking}

\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.
534 \def\MT@is@feature#1{%
535 \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
536 \ifMT@inlist@
537 \expandafter\@firstofone
538 \else
539 \MT@error{\@tempa' is not an available micro-typographic\MessageBreak
540 feature. Ignoring #1}{Available features are: \MT@features@long'.}%
541 \expandafter\@gobble
542 \fi
543 }

```

14.1.5 Compatibility

For the record, the following L^AT_EX kernel 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.

```

544 </package>
545 \let\MT@setup@\@empty

```

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

```

546 \def\MT@addto@setup{\g@addto@macro\MT@setup@}

```

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

```

547 \AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}

```

```

\MT@with@package We almost never do anything if a package is not loaded.
\MT@with@babel@and 548 \def\MT@with@package#1{\ifpackageloaded{#1}\@firstofone@gobble}
                    549 <*package>
                    550 \def\MT@with@babel@and#1{\ifpackagewith{babel}{#1}\@firstofone@gobble}

\MT@pdfcp@error Our competitor, the pdfcp@ package, must not be tolerated!
                    551 \def\MT@pdfcp@error{%
                    552   \MT@error{Detected the `pdfcp@' package!\MessageBreak
                    553     '\MT@MT' and `pdfcp@' may not be used together}{%
                    554   The `pdfcp@' package provides an interface to character protrusion.\MessageBreak
                    555   So does the '\MT@MT' package. Using both packages at the same\MessageBreak
                    556   time will almost certainly lead to undesired results. Have your choice!}%
                    557   \let\MT@pdfcp@error\relax
                    558 }
                    559 \MT@with@package{pdfcp@}\MT@pdfcp@error

Don't load letterspace.
                    560 \MT@let@nc{ver@letterspace.sty}\@empty

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

Peter Wilson was so kind to provide the \l@dunhbox@line hook in ledmac to
allow for protrusion. \leftmarginkern and \rightmarginkern are new primitives
of pdfTEX 1.21b (aka. 1.30.0).
                    561 \MT@requires@pdftex5{
                    562   \def\MT@ledmac@setup{%
                    563     \ifMT@protrusion
                    564       \MT@ifdefined@c@TF\l@dunhbox@line{%
                    565         \MT@info@n1{Patching ledmac to enable character protrusion}%
                    566         \newdimen\MT@led@kern
                    567         \let\MT@led@unhbox@line\l@dunhbox@line
                    568         \renewcommand*{\l@dunhbox@line}[1]{%
                    569           \ifhbox#1%
                    570             \MT@led@kern=\rightmarginkern##1%
                    571             \kern\leftmarginkern##1%
                    572             \MT@led@unhbox@line##1%
                    573             \kern\MT@led@kern
                    574           \fi
                    575         }%
                    576       }%
                    577       \MT@warning@n1{%
                    578         Character protrusion in paragraphs with line\MessageBreak
                    579         numbering will only work if you update ledmac}%
                    580     }%
                    581   \fi
                    582 }
                    583 }{
                    584   \def\MT@ledmac@setup{%
                    585     \ifMT@protrusion
                    586       \MT@warning@n1{%
                    587         The pdftex version you are using does not allow\MessageBreak
                    588         character protrusion in paragraphs with line\MessageBreak
                    589         numbering by the `ledmac' package.\MessageBreak
                    590         Upgrade pdftex to version 1.30 or later}%
                    591     \fi

```

```
592 }
593 }
```

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

```
594 \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`.)

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

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

```
596 \MT@with@babel@and{spanish}{\MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
```

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

```
597 \MT@with@package{csquotes}{%
```

```
598 \ifpackage@later{csquotes}{2005/05/11}\@disablequotes\relax}%
```

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

```
599 \ifpackage@loaded{hyperref}\MT@restore@p@h{%
```

```
600 \MT@with@package{tex4ht}\MT@restore@p@h}%
```

```
601 }
```

Check again at the end of the preamble.

```
602 </package>
```

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

```
604 <*package>
```

```
605 \MT@with@package{pdfcprot}\MT@pdfcprot@error
```

```
606 \MT@with@package{ledmac}\MT@ledmac@setup
```

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

```
607 \let\MT@setupfont@hook\@empty
```

```
608 \MT@with@babel@and{spanish}{%
```

```
609 \g@addto@macro\MT@setupfont@hook{%
```

```
610 \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
```

```
611 }
```

```
612 \MT@with@package{csquotes}{%
```

```
613 \ifpackage@later{csquotes}{2005/05/11}{%
```

```
614 \g@addto@macro\MT@setupfont@hook\@disablequotes
```

```
615 }{%
```

```
616 \MT@warning@n1{%
```

```
617 Should you receive warnings about unknown slot\MessageBreak
```

```
618 numbers, try upgrading the `csquotes' package}%
```

```
619 }%
```

```
620 }
```

```
621 </package>
```

```
622 \ifpackage@loaded{hyperref}{%
```

```
623 <package> \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
```

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

```

624 \pdfstringdefDisableCommands{%
625 <package> \let\pickup@font\MT@orig@pickupfont
626 \def\lststyle{\pdfstringdefWarn\lststyle}%
627 \def\textls#1#{\pdfstringdefWarn\textls}%
628 }%
629 }%
630 <*package>
631 \MT@with@package{tex4ht}{%
632 \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
633 }%
634 </package>
635 }
636 <*package>

```

The listings package makes numbers and letters active,

```

637 \MT@with@package{listings}{%
638 \g@addto@macro\MT@normal@catcodes{%
639 \MT@while@num{"30}{ "3A}{\catcode\@tempcnta 12\relax}%
640 \MT@while@num{"41}{ "5B}{\catcode\@tempcnta 11\relax}%
641 \MT@while@num{"61}{ "7B}{\catcode\@tempcnta 11\relax}%
642 }%

```

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

```

643 \g@addto@macro\MT@setupfont@hook{%
644 \catcode`\z@

```

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

```

645 \let\lst@ProcessLetter\@empty
646 }%
647 }

```

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.

```

648 %
649 </package>
650 \MT@with@package{soul}{%
651 \soulregister\lststyle 0%
652 \soulregister\textls 1%
653 }
654 <*package>

```

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

```

655 \MT@with@package{pinyin}{%
656 \let\MT@py@macron\py@macron
657 \ifpackagelater{pinyin}{2006/10/17}{% 4.7.0
658 \def\py@macron#1#2{%
659 \let\pickup@font\MT@orig@pickupfont
660 \MT@py@macron{#1}{#2}%
661 \let\pickup@font\MT@pickupfont}%
662 }%
663 \def\py@macron#1{%
664 \let\pickup@font\MT@orig@pickupfont
665 \MT@py@macron{#1}%
666 \let\pickup@font\MT@pickupfont}%
667 }%
668 }

```

```

669 </package>
670 }
671 <*package>

```

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

```

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

```

14.2 Font Setup

`\MT@setupfont` Setting up a font entails checking for each feature whether it should be applied to the current font (`\MT@font`).

```

673 \def\MT@setupfont{%

```

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

```

674 \MT@setupfont@hook

```

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

```

675 \MT@exp@two@c\MT@split@name\string\MT@font/\@nil

```

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

```

676 \MT@exp@one@n\MT@find@file\MT@family
677 \ifx\MT@familyalias\@empty \else
678 \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.)

```

679 \ifx\f@encoding\cf@encoding\else\@enc@update\fi

```

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

```

680 \MT@tracking
681 \MT@check@font
682 \ifMT@inlist@
683 <debug>\MT@show@pdfannot2%
684 \else
685 \MT@vinfo{Setting up font `~\MT@font'\on@line}%

```

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!

```

686 \MT@protrusion
687 \MT@expansion

```

Interword spacing and kerning.

```

688 \MT@spacing
689 \MT@kerning

```

Disable ligatures?

```

690 \MT@noligatures

```

Debugging.

```

691 <debug>\MT@show@pdfannot1%

```

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

```

692 \MT@register@font
693 \fi
694 \MT@gl@et\MT@font\@empty

```

```

695 }

\MT@split@name Split up the font name (<#6> may be the tracking amount).
\MT@encoding 696 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
\MT@family 697 \def\MT@encoding{#1}%
698 \def\MT@family {#2}%
\MT@series 699 \def\MT@series {#3}%
\MT@shape 700 \def\MT@shape {#4}%
\MT@size 701 \def\MT@size {#5}%

\MT@familyalias Alias family?
702 \MT@ifdefined@n@TF{MT@\MT@family @alias}%
703 {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
704 {\let\MT@familyalias\@empty}%
705 }

\ifMT@do We check all features of the current font against the lists of the currently active
\MT@feat font set, and set \ifMT@do accordingly.
\MT@maybe@do 706 \newif\ifMT@do
707 \def\MT@maybe@do#1{%

(but only if the feature isn't globally set to false)
708 \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).

709 \MT@dotrue
710 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
711 \MT@ifdefined@n@TF{MT@checklist@##1}%
712 {\csname MT@checklist@##1\endcsname}%
713 {\MT@checklist@{##1}}%
714 {#1}%
715 }%
716 \else
717 \MT@dofalse
718 \fi
719 \ifMT@do

\MT@feat stores the current feature.
720 \def\MT@feat{#1}%
721 \csname MT@set@#1@codes\endcsname
722 \else
723 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
724 \fi
725 }

\MT@checklist@ The generic test.
726 \def\MT@checklist@#1#2{%
727 \edef\@tempa{\csname MT@#2@setname\endcsname}%
728 !debug \MT@ifdefined@n@T
729 debug \MT@ifdefined@n@TF
730 {MT@#2list@#1@\@tempa}{%

Begin a \expandafter orgy to test whether the font attribute is in the list.
731 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
732 \csname MT@#1\expandafter\endcsname
733 \csname MT@#2list@#1@\@tempa\endcsname
734 \ifMT@inlist@
735 debug \MT@dinfo@n1{1}{\@nameuse{MT@abbr@#2}: #1 ` \@nameuse{MT@#1}' in list}%
736 \MT@dotrue

```

```

737     \else
738 <debug> \MT@dinfo{n}{1}{\@nameuse{MT@abbr@#2}: #1 ` \@nameuse{MT@#1}' not in list}%
739     \MT@dofalse
740     \expandafter\MT@clist@break
741     \fi
742 }%

```

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

```

743 <debug> {\MT@dinfo{n}{1}{\@nameuse{MT@abbr@#2}: #1 list empty}}%
744 }

```

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

```

745 \def\MT@checklist@font#1{%
746   \edef\@tempa{\csname MT@#1setname\endcsname}%
747   <!debug> \MT@ifdefined@n@T
748   <debug> \MT@ifdefined@n@TF
749   {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.

```

750   \expandafter\expandafter\expandafter\MT@exp@one@n
751   \expandafter\expandafter\expandafter\MT@in@clist
752   \expandafter\expandafter\expandafter{%
753     \expandafter\expandafter\expandafter\string
754     \expandafter\MT@font\expandafter}%
755   \csname MT@#1list@font@\@tempa\endcsname
756   \ifMT@inlist@
757   <debug> \MT@dinfo{n}{1}{\@nameuse{MT@abbr@#1}: font ` \MT@font' in list}%
758     \expandafter\MT@clist@break
759   \else
760   <debug> \MT@dinfo{n}{1}{\@nameuse{MT@abbr@#1}: font ` \MT@font' not in list}%
761   \MT@dofalse
762   \fi
763 }%
764 <debug> {\MT@dinfo{n}{1}{\@nameuse{MT@abbr@#1}: font list empty}}%
765 }

```

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

```

766 \def\MT@checklist@family#1{%
767   \edef\@tempa{\csname MT@#1setname\endcsname}%
768   <!debug> \MT@ifdefined@n@T
769   <debug> \MT@ifdefined@n@TF
770   {MT@#1list@family@\@tempa}{%
771     \MT@exp@two@n\MT@in@clist
772     \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
773     \ifMT@inlist@
774     <debug> \MT@dinfo{n}{1}{\@nameuse{MT@abbr@#1}: family ` \@nameuse{MT@family}' in list}%
775     \MT@dotrue
776   \else
777   <debug> \MT@dinfo{n}{1}{\@nameuse{MT@abbr@#1}: family ` \@nameuse{MT@family}' not in list}%
778   \MT@dofalse
779   \ifx\MT@familyalias\empty \else
780     \MT@exp@two@n\MT@in@clist
781     \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
782     \ifMT@inlist@
783     <debug> \MT@dinfo{n}{1}{\@nameuse{MT@abbr@#1}: alias ` \MT@familyalias' in list}%
784     \MT@dotrue
785   <debug> \else\MT@dinfo{n}{1}{\@nameuse{MT@abbr@#1}: alias ` \MT@familyalias' not in list}%
786   \fi

```



```

787     \fi
788     \fi
789     \ifMT@do \else
790         \expandafter\MT@clist@break
791     \fi
792 }%
793 (debug) {\MT@dinfo@n1}{\@nameuse{MT@abbr@#1}: family list empty}}%
794 }

```

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

```

795 \def\MT@checklist@size#1{%
796     \edef\@tempa{\csname MT@#1@setname\endcsname}%
797     (!debug) \MT@ifdefined@n@T
798     (debug) \MT@ifdefined@n@TF
799     {\MT@#1list@size@\@tempa}{%
800         \expandafter\MT@in@list
801         \csname MT@#1list@size@\@tempa\endcsname
802         \ifMT@in@list@
803     (debug) \MT@dinfo@n1}{\@nameuse{MT@abbr@#1}: size '\MT@size' in list}%
804         \MT@dotrue
805     \else
806     (debug) \MT@dinfo@n1}{\@nameuse{MT@abbr@#1}: size '\MT@size' not in list}%
807         \MT@dofalse
808         \expandafter\MT@clist@break
809     \fi
810 }%
811 (debug) {\MT@dinfo@n1}{\@nameuse{MT@abbr@#1}: size list empty}}%
812 }

```

14.2.1 Protrusion

`\MT@protrusion` Set up for protrusion?

```
813 \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.

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

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

```

815     \MT@if@list@exists{%
816         \MT@get@font@dimen@six
817         \MT@get@opt
818         \MT@reset@pr@codes

```

Get the name of the inheritance list and parse it.

```
819     \MT@get@inh@list
```

Set an input encoding?

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

Load additional lists?

```

821     \MT@load@list\MT@pr@c@name
822     \MT@set@listname

```

Load the main list.

```

823     \MT@let@cn\@tempc{\MT@pr@c@\MT@pr@c@name}%
824     \expandafter\MT@pr@do\@tempc,\relax,%
825 } \MT@reset@pr@codes
826 }

```

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

```
827 \def\MT@set@all@pr#1#2{%
828   (debug)\MT@info{n1}{3}{-- lp/rp: setting all to \number#1/\number#2}%
829   \MT@do@font{%
830     \lcode\MT@font\@tempcnta=#1\relax
831     \rcode\MT@font\@tempcnta=#2\relax
832   }%
833 }
```

`\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.

```
834 \def\MT@reset@pr@codes{\MT@set@all@pr\z@\z@}
835 \let\MT@reset@pr@codes\relax
```

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

`\MT@dimen@six`

`\MT@get@font@dimen@six`

```
836 \def\MT@get@font@dimen@six{%
837   \ifnum\fontdimen6\MT@font=\z@
838     \MT@warning{n1}{%
839       Font '\MT@font' does not specify its\MessageBreak
840       \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
841       \@nameuse{MT@abbr@\MT@feat} will not work with this font}%
842     \expandafter\MT@gobble@settings
843   \else
844     \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
845   \fi
846 }
847 \def\MT@gobble@settings#1\@tempc,\relax,{}%
```

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

```
848 \def\MT@pr@do#1,{%
849   \ifx\relax#1\@empty\else
850     \MT@pr@split #1==\relax
851     \expandafter\MT@pr@do
852   \fi
853 }
```

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

```
854 \def\MT@pr@split#1=#2=#3\relax{%
855   \def\@tempa{#1}%
856   \ifx\@tempa\@empty \else
857     \MT@get@slot
858     \ifnum\MT@char > \m@ne
859       \MT@get@char@unit
860       \MT@pr@split@val#2\relax
861     \fi
862   \fi
863 }
```

`\MT@pr@split@val`

```
864 \def\MT@pr@split@val#1,#2\relax{%
865   \def\@tempb{#1}%
866   \MT@ifempty\@tempb\relax{%
867     \MT@scale@to@em
868     \lcode\MT@font\MT@char=\@tempcntb
869   } (debug)\MT@info{n1}{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char: [#1]}%
870 }
```

```

871 \def\@tempb{#2}%
872 \MT@ifempty\@tempb\relax{%
873   \MT@scale@to@em
874   \rptcode\MT@font\MT@char=\@tempcntb
875 (debug)\MT@info{n1}{4}{;; rp (\MT@char): \number\rptcode\MT@font\MT@char: [#2]}%
876 }%

```

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

```

877 \MT@ifdefined@c@T\MT@pr@inh@name{%
878   \MT@ifdefined@n@T{\MT@inh@\MT@pr@inh@name @\MT@char @}{%
879     \expandafter\MT@map@tlist@c
880     \csname MT@inh@\MT@pr@inh@name @\MT@char @\endcsname
881     \MT@set@pr@heirs
882   }%
883 }%
884 }

```

`\MT@scale@to@em` Since pdf_T_EX 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 `\lptcode` resp. `\rptcode`, since this would disallow protrusion factors larger than the character width (since `\[1r]ptcode`’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`.

```

885 \MT@requires@pdftex3{
886   \def\MT@scale@to@em{%
887     \@tempcntb=\MT@count\relax

```

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

```

888   \MT@scale\@tempcntb \@tempb \MT@dimen@six
889   \ifnum\@tempcntb=\z@ \else
890     \MT@scale@factor
891   \fi
892 }

```

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

```

893 \MT@requires@etex{
894   \def\MT@get@charwd{%
895     \MT@count=\fontcharwd\MT@font\MT@char\relax
896     \ifnum\MT@count=\z@ \MT@info@missing@char \fi
897   }
898 }{
899   \def\MT@get@charwd{%
900     \setbox\z@=\hbox{\MT@font \char\MT@char}%
901     \MT@count=\wd\z@\relax
902     \ifnum\MT@count=\z@ \MT@info@missing@char \fi
903   }
904 }

```

No adjustment with versions 0.14f and 0.14g.

```

905 }{

```

```

906 \def\MT@scale@to@em{%
907   \MT@count=\@tempb\relax
908   \ifnum\MT@count=\z@ \else
909     \MT@scale@factor
910   \fi
911 }

```

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

```

912 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
913 }

```

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

```

914 \def\MT@get@font@dimen#1{%
915   \ifnum\fontdimen#1\MT@font=\z@
916     \MT@warning@nl{Font '\MT@font' does not specify its\MessageBreak
917       \@backslashchar fontdimen \number#1 (it's zero)!\MessageBreak
918       You should use a different `unit' for \MT@curr@list@name}%
919   \else
920     \MT@count=\fontdimen#1\MT@font
921   \fi
922 }

```

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

```

923 \MT@requires@etex{
924   \def\MT@info@missing@char{%
925     \MT@info@nl{Character '\the\MT@toks' \iffontchar\MT@font\MT@char
926       has a width of 0pt \else is missing \fi in font\MessageBreak
927       '\MT@font'. Ignoring protrusion settings\MessageBreak
928       for this character}%
929   }
930 }{
931   \def\MT@info@missing@char{%
932     \MT@info@nl{%
933       Character '\the\MT@toks' has a width of 0pt\MessageBreak
934       (it's probably missing) in font '\MT@font'.\MessageBreak
935       Ignoring protrusion settings for this character}%
936   }
937 }

```

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

```

938 \def\MT@scale@factor{%
939   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
940     \expandafter\MT@scale\expandafter \@tempcntb
941     \csname MT@\MT@feat @factor@\endcsname \@m
942   \fi
943   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
944     \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @max\endcsname
945   \else
946     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
947       \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @min\endcsname
948     \fi
949   \fi
950 }

```

\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.

```

951 \def\MT@warn@code@too@large#1{%
952   \@tempcnta=#1\relax
953   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
954     \expandafter\MT@scale\expandafter\@tempcnta\expandafter
955     \@m \csname MT@\MT@feat @factor@\endcsname

```

```

956 \fi
957 \MT@scale\@tempcnta \MT@dimen@six \MT@count
958 \MT@warning@n\{The \@nameuse{MT@abbr@MT@feat} code \@tempb\space
959 is too large for character\MessageBreak
960 ~\the\MT@toks' in \MT@curr@list@name.\MessageBreak
961 Setting it to the maximum of \number\@tempcnta}%
962 \@tempcntb=#1\relax
963 }

```

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

```

964 \def\MT@get@opt{%
965 \MT@set@listname

```

\MT@pr@factor@ Apply a factor?

```

\MT@sp@factor@ 966 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 967 \MT@let@nn{MT@\MT@feat @factor@}
968 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
969 \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@MT@feat} codes by
970 \number\csname MT@\MT@feat @factor@\endcsname/1000}%
971 }{%
972 \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
973 }%

```

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

```

\MT@sp@unit@ 974 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
\MT@kn@unit@ 975 \MT@let@nn{MT@\MT@feat @unit@}%
976 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
977 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\empty
978 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
979 relative to character widths}%
980 \else
981 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\m@ne
982 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
983 relative to width of space}%
984 \fi
985 \fi
986 }{%
987 \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
988 }%

```

\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 989 \let\MT@get@char@unit\relax
990 \let\MT@get@space@unit\gobble
991 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\empty
992 \let\MT@get@char@unit\MT@get@charwd
993 \else
994 \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\m@ne
995 \let\MT@get@space@unit\MT@get@font@dimen
996 \else
997 \expandafter\MT@get@unit\csname MT@\MT@feat @unit@\endcsname
998 \fi
999 \fi

```

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

```

1000 \MT@ifdefined@n@T{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
1001 \csname MT@preset@MT@feat\endcsname
1002 \MT@let@nc{MT@reset@MT@feat @codes}\relax

```

```
1003 }%
1004 }
```

`\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.

```
1005 \def\MT@get@unit#1{%
1006   \expandafter\MT@get@unit@#1 e!\@nil
1007   \ifx\x\@empty\else\let#1\x\fi
1008   \@defaultunits\@tempdima#1 pt\relax\@nnil
1009   \ifdim\@tempdima=\z@
1010     \MT@warning@n1{%
1011       Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
1012       width. Setting factors of list \@nameuse{MT@MT@feat @c@name}'\MessageBreak
1013       relative to character widths instead}%
1014     \let#1\@empty
1015     \let\MT@get@char@unit\MT@get@charwd
1016   \else
1017     \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
1018               to \the\@tempdima}%
1019     \MT@count=\@tempdima\relax
1020   \fi
1021 }
1022 \def\MT@get@unit@#1e#2#3\@nil{%
1023   \ifx\#3\\\let\x\@empty \else
1024     \if m#2%
1025       \edef\x{#1\fontdimen6\MT@font}%
1026     \else
1027       \if x#2%
1028         \edef\x{#1\fontdimen5\MT@font}%
1029       \fi
1030     \fi
1031   \fi
1032 }
```

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

```
1033 \def\MT@set@inputenc#1{%
```

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

```
1034   \def\MT@cat{#1}%
1035   \MT@ifdefined@n@T{%
1036     MT@MT@feat @#1\csname MT@MT@feat @#1\name\endcsname @inputenc}%
1037   \MT@ifdefined@c@TF\inputencoding{%
```

Set up normal catcodes, since, e. g., listings would otherwise want to typeset the `inputenc` file when it is loaded inside a listing.

```
1038     \MT@normal@catcodes
1039     <debug>\MT@dinfo@n1{1}{input encoding: \@nameuse{%
1040     <debug>MT@MT@feat @#1\csname MT@MT@feat @#1\name\endcsname @inputenc}}}%
1041     \inputencoding{\@nameuse{%
1042       MT@MT@feat @#1\csname MT@MT@feat @#1\name\endcsname @inputenc}}}%
1043   }{%
1044     \MT@error{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1045       \MessageBreak package isn't loaded. Ignoring key}%
1046     {You must load the `inputenc' package before you can use the `inputenc' key.}%
1047   }%
1048 }%
1049 }
```

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

```
1050 \def\MT@set@pr@heirs#1{%
1051   \lcode\MT@font#1=\lcode\MT@font\MT@char
1052   \rcode\MT@font#1=\rcode\MT@font\MT@char
1053   <debug>
1054   \MT@info@n1{2}{-- heir of \MT@char: #1}%
1055   \MT@info@n1{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
1056                                     \number\rcode\MT@font\MT@char}%
1057   </debug>
1058 }
```

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

```
\MT@preset@pr@ 1059 \def\MT@preset@pr{%
1060   \expandafter\expandafter\expandafter\MT@preset@pr@
1061   \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1062 }
1063 \def\MT@preset@pr#1,#2\@nil{%
1064   \ifx\MT@pr@unit@\@empty
1065     \MT@warn@preset@twidth{pr}%
1066     \let\MT@preset@aux\MT@preset@aux@factor
1067   \else
1068     \let\MT@preset@aux\MT@preset@aux@space
1069   \fi
1070   \MT@preset@aux{#1}\@tempa
1071   \MT@preset@aux{#2}\@tempb
1072   \MT@set@all@pr\@tempa\@tempb
1073 }
```

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

```
\MT@preset@aux@factor 1074 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 1075   \@tempcntb=#1\relax
1076   \MT@scale@factor
1077   \edef#2{\number\@tempcntb}%
1078 }
1079 \def\MT@preset@aux@space#1#2{%
1080   \def\@tempb{#1}%
1081   \MT@get@space@unit\tw@
1082   \MT@scale@to@em
1083   \edef#2{\number\@tempcntb}%
1084 }
```

`\MT@warn@preset@twidth`

```
1085 \def\MT@warn@preset@twidth#1{%
1086   \MT@warning@n1{%
1087     Cannot preset characters relative to their widths\MessageBreak
1088     for \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1089     \MessageBreak relative to lem instead}%
1090 }
```

14.2.2 Expansion

`\MT@expansion` Set up for expansion?

```
1091 \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).

```
1092 \def\MT@set@ex@codes@s{%
```

```

1093 \MT@if@list@exists{%
1094   \MT@get@ex@opt
1095   \MT@reset@ef@codes
1096   \MT@get@inh@list
1097   \MT@set@inputenc{c}%
1098   \MT@load@list\MT@ex@cname
1099   \MT@set@listname
1100   \MT@let@cn@tempc{\MT@ex@c@\MT@ex@cname}%
1101   \expandafter\MT@ex@do\tempc,\relax,%
1102   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1103 } \relax
1104 }

```

`\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.

```

1105 \newif\ifMT@nonselected
1106 \def\MT@set@ex@codes@n{%
1107   \MT@nonselectedtrue
1108   \MT@if@list@exists
1109   \MT@get@ex@opt
1110   {%
1111     \let\MT@stretch@\MT@stretch
1112     \let\MT@shrink@\MT@shrink
1113     \let\MT@step@\MT@step
1114     \let\MT@auto@\MT@auto
1115     \let\MT@ex@factor@\MT@ex@factor
1116   }%
1117   \MT@reset@ef@codes
1118   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1119   \MT@nonselectedfalse
1120 }

```

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

```
1121 \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@`

```

1122 \def\MT@set@all@ex#1{%
1123   <debug> \MT@dinfo{n}{3}{-- ex: setting all to \number#1}%
1124   \MT@do@font{\efcode\MT@font\tempcnta=#1\relax}%
1125 }
1126 \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.

```

1127 \MT@requires@pdfTeX4{
1128   \def\MT@reset@ef@codes{%
1129     \ifnum\MT@ex@factor@=\@m \else
1130       \MT@reset@ef@codes@
1131     \fi
1132   }
1133 }{
1134   \let\MT@reset@ef@codes\MT@reset@ef@codes@
1135 }

```

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

```

1136 \def\MT@ex@do#1,{%
1137   \ifx\relax#1\@empty \else
1138     \MT@ex@split #1==\relax

```



```

1139 \expandafter\MT@ex@do
1140 \fi
1141 }

```

\MT@ex@split

```

1142 \def\MT@ex@split#1=#2=#3\relax{%
1143 \def\@tempa{#1}%
1144 \ifx\@tempa\empty \else
1145 \MT@get@slot
1146 \ifnum\MT@char > \m@ne
1147 \@tempcntb=#2\relax

```

Take an optional factor into account.

```

1148 \ifnum\MT@ex@factor@=\@m \else
1149 \MT@scale\@tempcntb \MT@ex@factor@ \@m
1150 \fi
1151 \ifnum\@tempcntb > \MT@ex@max
1152 \MT@warn@ex@too@large\MT@ex@max
1153 \else
1154 \ifnum\@tempcntb < \MT@ex@min
1155 \MT@warn@ex@too@large\MT@ex@min
1156 \fi
1157 \fi
1158 \efcode\MT@font\MT@char=\@tempcntb
1159 <debug> \MT@info{n\{4\}::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#2]}%

```

Heirs, heirs, I love thy heirs.

```

1160 \MT@ifdefined@c\MT@ex@inh@name{%
1161 \MT@ifdefined@n\T\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1162 \expandafter\MT@map@tlist@c
1163 \csname \MT@inh@\MT@ex@inh@name @\MT@char @\endcsname
1164 \MT@set@ex@heirs
1165 }%
1166 }%
1167 \fi
1168 \fi
1169 }

```

\MT@warn@ex@too@large

```

1170 \def\MT@warn@ex@too@large#1{%
1171 \MT@warning{n\{Expansion factor \number\@tempcntb\space too large for
1172 character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1173 Setting it to the maximum of \number#1}%
1174 \@tempcntb=#1\relax
1175 }

```

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

```

\MT@ex@factor@ 1176 \def\MT@get@ex@opt{%
\MT@stretch@ 1177 \MT@set@listname
1178 \MT@ifdefined@n\TF\MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1179 \MT@let@cn\MT@ex@factor@\MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1180 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 1181 }{%
1182 \let\MT@ex@factor@\MT@ex@factor
1183 }%
1184 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1185 \MT@get@ex@opt@{shrink}{Setting shrink limit to \number\MT@shrink@}%
1186 \MT@get@ex@opt@{step}{Setting expansion step to \number\MT@step@}%
1187 \def\@tempa{autoexpand}%
1188 \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1189 \MT@ifdefined@n\T\MT@ex@c@\MT@ex@c@name @preset}{%
1190 \MT@preset@ex

```

```

1191 \let\MT@reset@ef@codes\relax
1192 }%
1193 }

\MT@get@ex@opt@
1194 \def\MT@get@ex@opt@#1#2{%
1195 \MT@ifdefined@n@TF{MT@ex@c@MT@ex@c@name @#1}{%
1196 \MT@let@nn{MT@#1@}{MT@ex@c@MT@ex@c@name @#1}%
1197 \MT@vinfo{... : #2}%
1198 }{%
1199 \MT@let@nn{MT@#1@}{MT@#1}%
1200 }%
1201 }

\MT@set@ex@heirs
1202 \def\MT@set@ex@heirs#1{%
1203 \efcode\MT@font#1=\efcode\MT@font\MT@char
1204 *debug
1205 \MT@dinfo@n1{2}{-- heir of \MT@char: #1}%
1206 \MT@dinfo@n4{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1207 /debug
1208 }

\MT@preset@ex
1209 \def\MT@preset@ex{%
1210 \@tempcntb=\csname MT@ex@c@MT@ex@c@name @preset\endcsname\relax
1211 \MT@scale@factor
1212 \MT@set@all@ex\@tempcntb
1213 }

```

14.2.3 Interword Spacing (Glue)

```

\MT@spacing Adjustment of interword spacing?
1214 \MT@requires@pdftex6{
1215 \def\MT@spacing{\MT@maybe@do{sp}}

\MT@set@sp@codes This is all the same.
1216 \def\MT@set@sp@codes{%
1217 \MT@if@list@exists{%
1218 \MT@get@font@dimen@six
1219 \MT@get@opt
1220 \MT@reset@sp@codes
1221 \MT@get@inh@list
1222 \MT@set@inputenc{c}%
1223 \MT@load@list\MT@sp@c@name
1224 \MT@set@listname
1225 \MT@let@cn\@tempc{MT@sp@c@MT@sp@c@name}%
1226 \expandafter\MT@sp@do\@tempc,\relax,%
1227 }\MT@reset@sp@codes
1228 }

\MT@sp@do
1229 \def\MT@sp@do#1,{%
1230 \ifx\relax#1\@empty \else
1231 \MT@sp@split #1==\relax
1232 \expandafter\MT@sp@do
1233 \fi
1234 }

\MT@sp@split

```

```

1235 \def\MT@sp@split#1=#2=#3\relax{%
1236   \def\@tempa{#1}%
1237   \ifx\@tempa\empty \else
1238     \MT@get@slot
1239     \ifnum\MT@char > \m@ne
1240       \MT@get@char@unit
1241       \MT@sp@split@val#2\relax
1242     \fi
1243   \fi
1244 }

```

\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).

```

1245 \def\MT@sp@split@val#1,#2,#3\relax{%
1246   \def\@tempb{#1}%
1247   \MT@ifempty\@tempb\relax{%
1248     \MT@get@space@unit\tw@
1249     \MT@scale@to@em
1250     \knbscode\MT@font\MT@char=\@tempcntb
1251     (debug)\MT@dinfo@n1{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1252   }%
1253   \def\@tempb{#2}%
1254   \MT@ifempty\@tempb\relax{%
1255     \MT@get@space@unit\thr@@
1256     \MT@scale@to@em
1257     \stbscode\MT@font\MT@char=\@tempcntb
1258     (debug)\MT@dinfo@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1259   }%
1260   \def\@tempb{#3}%
1261   \MT@ifempty\@tempb\relax{%
1262     \MT@get@space@unit4%
1263     \MT@scale@to@em
1264     \shbscode\MT@font\MT@char=\@tempcntb
1265     (debug)\MT@dinfo@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1266   }%
1267   \MT@ifdefined@c@T\MT@sp@inh@name{%
1268     \MT@ifdefined@n@T\MT@inh@\MT@sp@inh@name @\MT@char @}{%
1269       \expandafter\MT@map@tlist@c
1270       \csname MT@inh@\MT@sp@inh@name @\MT@char @\endcsname
1271       \MT@set@sp@heirs
1272     }%
1273   }%
1274 }

```

\MT@set@sp@heirs

```

1275 \def\MT@set@sp@heirs#1{%
1276   \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1277   \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1278   \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1279   (*debug)
1280   \MT@dinfo@n1{2}{-- heir of \MT@char: #1}%
1281   \MT@dinfo@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1282                                     \number\stbscode\MT@font\MT@char/%
1283                                     \number\shbscode\MT@font\MT@char}%
1284   (/debug)
1285 }

```

\MT@set@all@sp

```

\MT@reset@sp@codes 1286 \def\MT@set@all@sp#1#2#3{%
1287 (debug)\MT@dinfo@n1{3}{-- knbs/stbs/shbs: setting all to \number#1/\number#2/\number#3}%
\MT@reset@sp@codes@ 1288 \MT@do@font{%
1289   \knbscode\MT@font\@tempcnta=#1\relax

```

```

1290 \stbrcode\MT@font\@tempcnta=#2\relax
1291 \shbrcode\MT@font\@tempcnta=#3\relax
1292 }%
1293 }
1294 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1295 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1296 \def\MT@preset@sp{%
1297 \expandafter\expandafter\expandafter\MT@preset@sp@
1298 \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1299 }
1300 \def\MT@preset@sp@#1,#2,#3\@nil{%
1301 \ifx\MT@sp@unit@\@empty
1302 \MT@warn@preset@t@width{sp}%
1303 \MT@preset@aux@factor{#1}\@tempa
1304 \MT@preset@aux@factor{#2}\@tempc
1305 \MT@preset@aux@factor{#3}\@tempb
1306 \else
1307 \MT@preset@aux@space{#1}\@tempa
1308 \def\@tempb{#2}%
1309 \MT@get@space@unit\thr@@
1310 \MT@scale@to@em
1311 \edef\@tempc{\number\@tempcntb}%
1312 \def\@tempb{#3}%
1313 \MT@get@space@unit4%
1314 \MT@scale@to@em
1315 \edef\@tempb{\number\@tempcntb}%
1316 \fi
1317 \MT@set@all@sp\@tempa\@tempc\@tempb
1318 }

Only for sufficiently new versions of pdfTeX.
1319 }{
1320 \let\MT@spacing\relax
1321 }

```

14.2.4 Additional Kerning

```

\MT@kerning Again, only check for additional kerning for new versions of pdfTeX.
1322 \MT@requires@pdftex6{
1323 \def\MT@kerning{\MT@maybe@do{kn}}

\MT@set@kn@codes It's getting boring, I know.
1324 \def\MT@set@kn@codes{%
1325 \MT@if@list@exists{%
1326 \MT@get@font@dimen@six
1327 \MT@get@opt
1328 \MT@reset@kn@codes
1329 \MT@get@inh@list
1330 \MT@set@inputenc{c}%
1331 \MT@load@list\MT@kn@c@name
1332 \MT@set@listname
1333 \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
1334 \expandafter\MT@kn@do\@tempc,\relax,%
1335 }\MT@reset@kn@codes
1336 }

\MT@kn@do
1337 \def\MT@kn@do#1,{%
1338 \ifx\relax#1\@empty \else

```

```

1339 \MT@kn@split #1==\relax
1340 \expandafter\MT@kn@do
1341 \fi
1342 }

```

\MT@kn@split

```

1343 \def\MT@kn@split#1=#2=#3\relax{%
1344 \def\@tempa{#1}%
1345 \ifx\@tempa\empty \else
1346 \MT@get@slot
1347 \ifnum\MT@char > \m@ne
1348 \MT@get@char@unit
1349 \MT@kn@split@val#2\relax
1350 \fi
1351 \fi
1352 }

```

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

```

1353 \def\MT@kn@split@val#1,#2\relax{%
1354 \def\@tempb{#1}%
1355 \MT@ifempty\@tempb\relax{%
1356 \MT@get@space@unit\tw@
1357 \MT@scale@to@em
1358 \knbcode\MT@font\MT@char=\@tempcntb
1359 <debug>\MT@info{n1}{4}{;;; knbc (\MT@char): \number\knbcode\MT@font\MT@char: [#1]}%
1360 }%
1361 \def\@tempb{#2}%
1362 \MT@ifempty\@tempb\relax{%
1363 \MT@get@space@unit\tw@
1364 \MT@scale@to@em
1365 \knaccode\MT@font\MT@char=\@tempcntb
1366 <debug>\MT@info{n1}{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1367 }%
1368 \MT@ifdefined@c@T\MT@kn@inh@name{%
1369 \MT@ifdefined@n@T{\MT@inh@\MT@kn@inh@name @\MT@char @}{%
1370 \expandafter\MT@map@tlist@c
1371 \csname MT@inh@\MT@kn@inh@name @\MT@char @\endcsname
1372 \MT@set@kn@heirs
1373 }%
1374 }%
1375 }

```

\MT@set@kn@heirs

```

1376 \def\MT@set@kn@heirs#1{%
1377 \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1378 \knaccode\MT@font#1=\knaccode\MT@font\MT@char
1379 <*debug>
1380 \MT@info{n1}{2}{-- heir of \MT@char: #1}%
1381 \MT@info{n1}{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1382 \number\knaccode\MT@font\MT@char}%
1383 </debug>
1384 }

```

\MT@set@all@kn

```

\MT@reset@kn@codes 1385 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1386 <debug>\MT@info{n1}{3}{-- knac/knbc: setting all to \number#1/\number#2}%
1387 \MT@do@font{%
1388 \knbcode\MT@font\@tempcnta=#1\relax
1389 \knaccode\MT@font\@tempcnta=#2\relax
1390 }%
1391 }
1392 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}

```

```

1393 \let\MT@reset@kn@codes\relax

\MT@preset@kn
\MT@preset@kn@ 1394 \def\MT@preset@kn{%
1395   \expandafter\expandafter\expandafter\MT@preset@kn@
1396   \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1397 }
1398 \def\MT@preset@kn@#1,#2\@nil{%
1399   \ifx\MT@kn@unit@\@empty
1400     \MT@warn@preset@tewidth{kn}%
1401     \let\MT@preset@aux\MT@preset@aux@factor
1402   \else
1403     \let\MT@preset@aux\MT@preset@aux@space
1404   \fi
1405   \MT@preset@aux{#1}\@tempa
1406   \MT@preset@aux{#2}\@tempb
1407   \MT@set@all@kn\@tempa\@tempb
1408 }
1409 }{
1410   \let\MT@kerning\relax
1411 }
1412 </package>

```

14.2.5 Tracking

`\lsstyle` This only works with pdfTeX 1.40.

```

1413 <package> \MT@requires@pdftex6{
1414   \DeclareRobustCommand\lsstyle{%
1415     <*package>
1416     \MT@trackingtrue
1417     \def\MT@tr@setname{all}%
1418   </package>
1419   \let\MT@tracking\MT@tracking@

```

Enable the (protected) `\lslig` command.

```

1420   \MT@let@nc{lslig}\MT@lslig
1421   \selectfont

```

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

```

1422   \MT@lskern=\dimexpr\MT@ifdefined@c@TF\MT@letterspace@
1423   \MT@letterspace@\MT@letterspace sp
1424   * \fontdimen6\MT@lsfont/2000\relax
1425   \MT@ls@adjust
1426 }

```

`\MT@tracking` We do not check whether we’ve already seen the font!

```

\MT@tracking@ 1427 \def\MT@tracking@{%
1428   <package> \ifMT@tracking\MT@maybe@do{tr}\fi
1429   <letterspace> \MT@set@tr@codes
1430 }
1431 <package> \let\MT@tracking\MT@tracking@
1432 <letterspace> \let\MT@tracking\relax

```

`\MT@set@tr@codes` The tracking amount is determined by the optional argument to `\textls`, settings from `\SetTracking`, or the global `letterspace` option, in this order.

```

1433 \def\MT@set@tr@codes{%
1434   \MT@ifdefined@c@TF\MT@letterspace@\relax{%
1435     \let\MT@letterspace@\MT@letterspace
1436   <*package>

```

```

1437 \MT@if@list@exists{%
1438 \MT@set@listname
1439 \MT@let@cn\MT@letterspace@{MT@tr@cc@MT@tr@cc@name}%

```

Different unit?

```

1440 \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name @unit}{%
1441 \MT@let@cn\MT@tr@unit@{MT@tr@cc@MT@tr@cc@name @unit}%
1442 \ifdim\MT@tr@unit@=1em\else
1443 \MT@get@font@dimen@six
1444 \MT@let@cn\@tempb{MT@tr@cc@MT@tr@cc@name}%
1445 \MT@get@unit\MT@tr@unit@
1446 \let\MT@tr@factor@\@m
1447 \MT@scale@to@em
1448 \edef\MT@letterspace@{\number\@tempcntb}%
1449 \fi
1450 }%
1451 }\relax
1452 </package>
1453 }%
1454 \ifnum\MT@letterspace@=\z@&else

```

`\MT@lsfont` The letterspaced font instances are saved in macros `\font name)/letterspacing amount)`ls.

```

1455 \xdef\MT@lsfont{%
1456 <package> \expandafter\csname\expandafter\string\MT@font
1457 <letterspace> \csname\curr@fontshape\fontsize
1458 /number\MT@letterspace@ 1s\endcsname}%
1459 \expandafter\ifx\MT@lsfont\relax

```

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

```

1460 \MT@get@ls@basefont
1461 <debug>\MT@dinfo@n1{1}{... new letterspacing instance}%
1462 \global\expandafter\letterspacefont\MT@lsfont\font@name \MT@letterspace@

```

Scale interword spacing (`\fontdimen 2`).

```

1463 \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1464 * \fontdimen2\MT@lsfont/1000\relax
1465 \fi
1466 <*package>
1467 \MT@vinfo{Tracking font '\MT@@font' by \number\MT@letterspace@
1468 \space\on@line}%
1469 \let\MT@font\MT@lsfont
1470 \aftergroup
1471 </package>
1472 \MT@set@lsfont
1473 \fi
1474 }

```

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

```

1475 \def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}

```

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

```

1476 \DeclareRobustCommand\lslig{\@firstofone}
1477 \def\MT@lslig#1{%
1478 \MT@get@ls@basefont
1479 \kern\MT@lskern{\font@name #1}\kern\MT@lskern
1480 }

```

`\MT@get@ls@basefont` pdf_T_EX cannot letterspace fonts that are already letterspaced. Therefore, we have to extract the base font, if we find that `\font@name` contains a ‘+’ or a ‘-’ (with `\pdfmatch` bullet-proof even if the font name contains one of these letters).

```

1481 \def\MT@get@ls@basefont{%
1482   \ifcase\pdfmatch{^select font (.*)[+-][0-9]*ls( at .*)?}$}
1483     {\expandafter\meaning\font@name}\relax
1484   \else
1485     <debug>\MT@dinfo@n1{1}{... fixing base font}%
1486     \global\expandafter\font\font@name
1487     \expandafter\strip@prefix\pdf@lastmatch1
1488     \expandafter\strip@prefix\pdf@lastmatch2\relax
1489   \fi
1490 }
1491 <*package>

```

For older pdf_T_EX versions, let it relax.

```

1492 }{
1493   \let\MT@tracking\relax
1494   \DeclareRobustCommand\lsstyle{%
1495     \MT@warning{Letterspacing only works with pdftex version 1.40\MessageBreak
1496       or newer. You might want to use the `soul' package\MessageBreak
1497       instead}%
1498     \MT@glet\lsstyle\relax
1499   }
1500 }
1501 </package>

```

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

```

1502 \DeclareRobustCommand\textls{%
1503   \hmode\bgroup
1504   \@ifstar{\let\MT@ls@adjust\MT@ls@adjust@MT@textls}%
1505     {\let\MT@ls@adjust\relax \MT@textls}%
1506 }

```

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

```

\MT@letterspace@ 1507 \newcommand\MT@textls[2] [] {%
1508   \MT@ifempty{#1}%
1509     {\let\MT@letterspace@ \@undefined}%
1510     {\KV@esp@def\MT@letterspace@{#1}%
1511       \MT@ls@too@large\MT@letterspace@}%
1512   \lsstyle #2%
1513   \MT@ls@adjust
1514   \egroup
1515 }

```

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

```

1516 \def\MT@ls@too@large#1{%
1517   \ifnum#1>\MT@tr@max
1518     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
1519     \let#1\MT@tr@max
1520   \else
1521     \ifnum#1<\MT@tr@min
1522       \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
1523       \let#1\MT@tr@min
1524     \fi
1525   \fi
1526 }

```

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

```

1527 \newdimen\MT@lskern

```


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

```
\MT@ls@adjust@ 1528 \let\MT@ls@adjust\relax
1529 \def\MT@ls@adjust@{\kern-\MT@lskern}
1530 <package>
```

14.2.6 Disabling Ligatures

`\MT@noligatures` The possibility to disable ligatures is a new features of pdf_{TeX} 1.30.

```
1531 \MT@requires@pdftex5{
1532   \def\MT@noligatures{%
1533     \MT@dotrue
1534     \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1535       \MT@ifdefined@n@TF{MT@checklist@##1}%
1536       {\csname MT@checklist@##1\endcsname}%
1537       {\MT@checklist@{##1}}%
1538     }%
1539   }%
1540   \ifMT@do
1541     \pdfnoligatures\MT@font
1542     \MT@vinfo{... Disabling ligatures}%
1543   \fi
1544 }
1545 {}
1546 \let\MT@noligatures\relax
1547 }
```

14.2.7 Loading the Configuration

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

```
1548 \def\MT@load@list#1{%
1549   \edef\@tempa{#1}%
1550   \MT@let@cn\@tempb{MT@\MT@feat @c@\@tempa @load}%
1551   \MT@ifstreq\@tempa\@tempb{%
1552     \MT@warning{\@nameuse{MT@abbr@\MT@feat} list '\@tempa' cannot load itself}%
1553   }{%
1554     \ifx\@tempb\relax \else
1555       \MT@ifdefined@n@TF{MT@\MT@feat @c@\@tempb}{%
1556         \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list '\@tempb'}%
1557         \begingroup
1558           \MT@load@list\@tempb
1559         \endgroup
1560         \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list
1561           \noexpand\MessageBreak'\@tempb'}%
1562         \MT@let@cn\@tempc{MT@\MT@feat @c@\@tempb}%
1563         \expandafter\csname MT@\MT@feat @do\expandafter\endcsname\@tempc,\relax,%
1564       }{%
1565         \MT@warning{\@nameuse{MT@abbr@\MT@feat} list '\@tempb' undefined.
1566           Cannot load\MessageBreak it from list '\@tempa'}%
1567       }%
1568     \fi
1569   }%
1570 }
```

`\MT@find@file` Micro-typographic settings may be written into a file `mt-.cfg`.

`\MT@file@list` We must also record whether we've already loaded the file.

```
1571 \let\MT@file@list\empty
1572 \def\MT@find@file#1{%
```

Check for existence of the file only once.

```
1573 \MT@in@clist{#1}\MT@file@list
1574 \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.

```
1575 \MT@begin@catcodes
1576 \let\MT@begin@catcodes\relax
1577 \let\MT@end@catcodes\relax
1578 \InputIfFileExists{mt-#1.cfg}{%
1579 \edef\MT@curr@file{mt-#1.cfg}%
1580 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1581 \MT@xadd\MT@file@list{#1,}%
1582 }{%
1583 \expandafter\MT@get@basefamily#1\relax\relax\relax
1584 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1585 \ifMT@inlist@\else
1586 \InputIfFileExists{mt-\@tempa.cfg}{%
1587 \edef\MT@curr@file{mt-\@tempa.cfg}%
1588 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1589 \MT@xadd\MT@file@list{\@tempa,#1,}%
1590 }{%
1591 \MT@vinfo{... No configuration file mt-#1.cfg}%
1592 \MT@xadd\MT@file@list{#1,}%
1593 }%
1594 \fi
1595 }%
1596 \endgroup
1597 \fi
1598 }
```

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

OK, now all printable characters up to 127 are 'other'. We hope that letters are always letters and numbers other. (`\listings` makes them active, see section 14.1.5.)

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

```
1599 \def\MT@normal@catcodes{%
1600 \makeatletter
1601 \catcode`\^7%
1602 \catcode`\_9%
1603 \catcode`\^^I9%
1604 \catcode`\^^M9%
1605 \catcode`\z@
1606 \catcode`\{@ne
1607 \catcode`\}\tw@
1608 \catcode`\#6%
1609 \catcode`\%14%
1610 \MT@map@tlist@n
1611 {\!\\"$&\'(\)\*\+\,\-\.\/\:\;<=\>?\[\]\_-\`\/\~}%
1612 \@makeother
1613 }
```

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

```
1614 \def\MT@begin@catcodes{%
1615 \begingroup
```

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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```
1616 \MT@normal@catcodes
```

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

```
1617 \let\KV@@sp@def\def
```

```
1618 }
```

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

```
1619 \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 ...).

```
1620 \def\MT@get@basefamily#1#2#3#4\relax{%
1621   \ifx#2\relax \def\@tempa{#1}\else
1622     \ifx#3\relax \def\@tempa{#1#2}\else
1623       \def\@tempa{#1#2#3}%
1624     \ifx\relax#4\relax \else
1625       \MT@ifstreq{#4}{\string x}\relax{%
1626         \MT@ifstreq{#4}{\string j}\relax{%
1627           \MT@ifstreq{#4}{\string w}\relax{%
1628             \def\@tempa{#1#2#3#4}}}\fi\fi\fi
1629 }
```

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

`\MT@get@listname`

```
\MT@get@listname@ 1630 \def\MT@get@listname#1{%
1631   (debug) \MT@dinfo{n1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@@font'}%
1632   \let\MT@listname\undefined
1633   \def\@tempb{#1}%
1634   \MT@map@tlist@c\MT@try@order\MT@get@listname@
1635 }
1636 \def\MT@get@listname@#1{%
1637   \expandafter\MT@next@listname#1%
1638   \ifx\MT@listname\undefined \else
1639     \expandafter\MT@tlist@break
1640   \fi
1641 }
```

`\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.

```
1642 \def\MT@try@order{%
1643   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
1644   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
1645 }
```

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

```
1646 \def\MT@next@listname#1#2#3#4{%
1647   \edef\@tempa{\MT@encoding
```

```

1648         /ifnum#1=\@ne \MT@family\fi
1649         /ifnum#2=\@ne \MT@series\fi
1650         /ifnum#3=\@ne \MT@shape\fi
1651         /ifnum#4=\@ne *\fi
1652         \MT@context}%
1653 <debug>\MT@info@n1{1}{trying \@tempa}%
1654 \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
1655     \MT@next@listname@#4%
1656 }{%

```

Also try with an alias family.

```

1657     \ifnum#1=\@ne
1658     \ifx\MT@familyalias\empty \else
1659     \edef\@tempa{\MT@encoding
1660         /MT@familyalias
1661         /ifnum#2=\@ne \MT@series\fi
1662         /ifnum#3=\@ne \MT@shape\fi
1663         /ifnum#4=\@ne *\fi
1664         \MT@context}%
1665 <debug>\MT@info@n1{1}{(alias) \@tempa}%
1666 \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
1667     \MT@next@listname@#4%
1668 }%
1669     \fi
1670 \fi
1671 }%
1672 }

```

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

```

1673 \def\MT@next@listname@#1{%
1674     \ifnum#1=\@ne
1675     \expandafter\MT@in@rlist\csname MT@\@tempb @\@tempa @size\endcsname
1676     \ifMT@inlist@
1677     \let\MT@listname\MT@size@name
1678     \fi
1679     \else
1680     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
1681     \fi
1682 }

```

\MT@if@list@exists

```

\MT@context 1683 \def\MT@if@list@exists{%
1684     \MT@let@cn\MT@context{MT@\MT@feat @context}%
1685     \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
1686     \MT@get@listname{\MT@feat @c}%
1687     \MT@ifdefined@c@TF{MT@listname}%
1688     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
1689     \ifMT@nonselected
1690     \MT@vinfo{... Applying non-selected expansion (list '\MT@listname')}%
1691     \else
1692     \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list '\MT@listname'}%
1693     \fi
1694     \@firstoftwo
1695 }{%

```

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

```

1696     \MT@let@nc{MT@\MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

1697     \ifMT@nonselected
1698     \MT@vinfo{... Applying non-selected expansion (no list)}%

```

```
1699 \else
```

Tracking doesn't require a list, either.

```
1700 \MT@ifstreq\MT@feat{tr}\relax{%
1701 \MT@warning[I cannot find a \@nameuse{MT@abbr@\MT@feat} list
1702 for font\MessageBreak`\MT@font'%
1703 \ifx\MT@context\@empty\else\space(context: `\MT@context')\fi.
1704 Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
1705 }%
1706 \fi
1707 \@secondoftwo
1708 }%
1709 }
```

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

```
\MT@context 1710 \def\MT@get@inh@list{%
1711 \let\MT@context\@empty
1712 \MT@get@listname{\MT@feat @inh}%
1713 \MT@ifdefined@c@TF\MT@listname{%
1714 \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
1715 (debug)
1716 \MT@dinfo@n{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
1717 \MT@listname'}%
1718 (debug)
1719 \MT@let@cn@tempc{MT@\MT@feat @inh@\MT@listname}%
```

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

```
1720 \ifx\@tempc\@empty \else
1721 (debug)\MT@dinfo@n{1}{parsing inheritance list ...}%
```

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

```
1722 \begingroup
1723 \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\MT@listname'}%
1724 \MT@set@inputenc{inh}%
1725 \expandafter\MT@inh@do\@tempc,\relax,%
1726 \global\MT@let@nc{MT@\MT@feat @inh@\MT@listname}\@empty
1727 \endgroup
1728 \fi
1729 }{%
1730 \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
1731 }%
1732 }
```

14.2.8 Translating Characters into Slots

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

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

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

```
\MT@char@ 1733 \def\MT@get@slot{%
1734 \escapechar`\
1735 \let\MT@char@\m@ne
1736 \MT@noesttrue
```

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

```
1737 \MT@toks=\expandafter{\@tempa}%
```

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

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

```
1738      \expandafter\MT@is@letter\@tempa\relax\relax
1739      \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.

```
1740      \MT@exp@two@c\MT@is@active\string\@tempa\@nil
```

- OK, so it must be a macro. We do not allow random commands but only those defined in L^AT_EX's idiosyncratic font encoding scheme:

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

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

```
1741      \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
1742      \MT@is@symbol
```

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

```
1743      {\expandafter\MT@is@composite\@tempa\relax\relax}%
1744      \ifnum\MT@char@ < \z@
```

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

```
1745      \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1746      \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
1747      \fi
1748      \fi

1749      \let\MT@char\MT@char@
1750      \ifnum\MT@char < \z@
1751      \MT@warn@unknown
1752      \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.

```
1753      \ifMT@noreset \else
1754      \MT@warn@reset
1755      \let\MT@char@m@ne
1756      \fi
1757      \fi
1758      \escapechar@m@ne
1759      }
```

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

```
\MT@testreset 1760 \newif\ifMT@noreset
1761 \def\MT@testreset#1#2{\MT@ifstreq{#1}{#2}\relax\MT@noresetfalse}
```

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

```
1762 \def\MT@is@letter#1#2\relax{%
1763   \ifcat a\noexpand#1\relax
1764   \edef\MT@char@\{number`#1\}%
```

```

1765 \ifx\#2\%
1766 <debug>\MT@info{n}{3}{> `the\MT@toks' is a letter (\MT@char@)}%
1767 \else
1768 \MT@norestfalse
1769 \fi
1770 \else
1771 \ifcat !\noexpand#1\relax
1772 \edef\MT@char@{\number`#1}%
1773 <debug>\MT@info{n}{3}{> `the\MT@toks' is a character (\MT@char@)}%
1774 \ifx\#2\%
1775 \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
1776 \else
1777 \MT@norestfalse
1778 \expandafter\MT@is@number#1#2\relax\relax
1779 \fi
1780 \fi
1781 \fi
1782 }

```

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

```

1783 \def\MT@is@number#1#2#3\relax{%
1784 \ifx\relax#3\relax \else
1785 \ifx\relax#2\relax \else
1786 \MT@noresttrue
1787 \if#1"\relax
1788 \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
1789 <debug>\MT@info{n}{3}{> ... a hexadecimal number: \MT@char@}%
1790 \else
1791 \if#1'\relax
1792 \def\MT@char@{\number#1#2#3}%
1793 <debug>\MT@info{n}{3}{> ... an octal number: \MT@char@}%
1794 \else
1795 \MT@ifint{#1#2#3}{%
1796 \def\MT@char@{\number#1#2#3}%
1797 <debug>\MT@info{n}{3}{> ... a decimal number: \MT@char@}%
1798 } \MT@norestfalse
1799 \fi
1800 \fi
1801 \ifnum\MT@char@ > \ccclv
1802 \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
1803 \let\MT@char@\m@ne
1804 \fi
1805 \fi
1806 \fi
1807 }

```

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

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

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

inputenc/utf8x).

```
1808 \def\MT@is@active#1#2\@nil{%
1809   \ifnum\catcode`#1 = \active
1810     \begingroup
1811       \set@display@protect
1812       \let\IeC\@firstofone
1813       \let\@inpen@undefined@MT@undefined@char
```

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

```
1814   \def\UTFviii@defined#1{\ifx #1\relax
1815     \MT@undefined@char{utf8}\else\expandafter #1\fi}%
1816   \edef\x{%
1817     \def\noexpand\@tempa{\@tempa}%
```

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

```
1818     \MT@toks={\the\MT@toks\space(= \@tempa)}%
1819   }%
1820   \expandafter\endgroup\x
1821 \fi
1822 }
```

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

```
1823 \def\MT@undefined@char#1{undefined in input encoding ``#1''}
```

\MT@is@symbol The symbol commands might expand to funny stuff, depending on context. Instead of simply expanding `\<command>`, we construct the command `\<encoding>\<command>` and see whether its meaning is `\char"<hex number>`, which is the case for everything that has been defined with `\DeclareTextSymbol` in the encoding definition files.

```
1824 \def\MT@is@symbol{%
1825   \expandafter\def\expandafter\MT@char\expandafter
1826     {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
1827   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1828     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
1829   \ifnum\MT@char@ < \z@
```

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

```
1830   \expandafter\MT@is@letter\MT@char\relax\relax
1831 \fi
1832 }
```

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

```
\MT@charstring 1833 \begingroup
1834   \catcode`\=/0
1835   /MT@map@tlist@n{/CHAR}/@makeother
1836   /lowercase{%
1837     /def/x{%
1838       /def/MT@charstring{CHAR"%
1839       /def/MT@is@char##1CHAR"##2##3##4/relax{%
1840         /ifx/relax##1/relax
1841         /if##3\relax
1842         /edef/MT@char@{/number"##2}%
1843         /MT@testrest/MT@charstring{##3##4}%
1844         /else
1845         /edef/MT@char@{/number"##2##3}%
1846         /MT@testrest/MT@charstring{##4}%
1847         /fi
1848         (debug) /MT@edinfo@n1{3}{> `the/MT@toks' is a \char (/MT@char@)}%
1849         /fi
```



```

1850     }%
1851   }%
1852 }
1853 /expandafter/endgroup/x

```

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

```

1854 \def\MT@is@composite#1#2\relax{%
1855   \ifx\#2\\else

```

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

```

1856   \edef\MT@char{\expandafter\csname\expandafter
1857                 \string\csname\MT@encoding\endcsname
1858                 \MT@detokenize@{#1}~%
1859                 \MT@detokenize@{#2}%
1860                 \endcsname}%
1861   \expandafter\MT@is@letter\MT@char\relax\relax
1862   \fi
1863 }

```

(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 1864 \def\MT@set@listname{%
1865   \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list\noexpand\MessageBreak
1866   ~\@nameuse{MT@MT@feat @c@name}}}%
1867 }

```

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

```

1868 \def\MT@warn@ascii{%
1869   \MT@warning@nl{Character `the\MT@toks' (= \MT@char@)
1870   is outside of ASCII range.\MessageBreak
1871   You must load the 'inputenc' package before using\MessageBreak
1872   8-bit characters in \MT@curr@list@name}%
1873 }

```

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

```
1874 \def\MT@warn@number@too@large#1{%
1875   \MT@warning@nl{%
1876     Number #1 in encoding '\MT@encoding' too large!\MessageBreak
1877     Ignoring it in \MT@curr@list@name}%
1878 }
```

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

```
1879 \def\MT@warn@rest{%
1880   \MT@warning@nl{%
1881     Unknown slot number of character\MessageBreak`the\MT@toks'%
1882     \MT@warn@maybe@inputenc\MessageBreak
1883     in font encoding '\MT@encoding'.\MessageBreak
1884     Make sure it's a single character\MessageBreak
1885     (or a number) in \MT@curr@list@name}%
1886 }
```

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

```
1887 \def\MT@warn@unknown{%
1888   \MT@warning@nl{%
1889     Unknown slot number of character\MessageBreak`the\MT@toks'%
1890     \MT@warn@maybe@inputenc\MessageBreak
1891     in font encoding '\MT@encoding' in \MT@curr@list@name}%
1892 }
```

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

```
1893 \def\MT@warn@maybe@inputenc{%
1894   \MT@ifdefined@n@T
1895   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
1896   { (input encoding '\@nameuse
1897     {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')}%
1898 }
```

14.2.9 Hook into L^AT_EX's font selection

We append `\MT@setupfont` to `\pickup@font`, which is called by L^AT_EX 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`), `tifa`, and probably many more. Furthermore, we had to include a hack for the `IEEEtran` class which loads all fonts in the class file itself (to fine tune inter-word spacing). Then I learned that even my 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.

```
1899 \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.

```
1900 \let\MT@font\@empty
1901 <package>
1902 \MT@addto@setup{%
1903 <package>
1904   \g@addto@macro\do@subst@correction{%
1905     \xdef\MT@font{\csname \curr@fontshape/\f@size\endcsname}%
1906   }
1907 </package>
```

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

```
1908 \ifpackageloaded{CJK}{
1909   \ifpackageafter{CJK}{2006/10/17}
1910   {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}} % 4.7.0
1911   {\def\MT@orig@pickupfont{\ifundefined{CJK@plane}}}
1912   \g@addto@macro\MT@orig@pickupfont{%
1913     {\expandafter\ifx\font@name\relax\define@newfont\fi}%
1914     {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
1915       \define@newfont\else\xdef\font@name
1916         {\csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}
1917   }{
1918     \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
1919   }
1920   \ifx\pickup@font\MT@orig@pickupfont \else
1921     \MT@warning@nl{%
1922       Command \string\pickup@font\space is not defined as expected.%
1923 <package>
1924     \MessageBreak Double-check whether micro-typography is indeed\MessageBreak
1925     applied to the document.\MessageBreak (Hint: Turn on `verbose' mode)%
1926 </package>
1927   }
1928 \fi
```

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

```
1929 \g@addto@macro\pickup@font{\begingroup}
```

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

```
1930 \MT@with@package{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
1931 \endpackage
1932 \g@addto@macro\pickup@font{%
1933 \letterspace \MT@tracking
1934 \*package
1935 \*debug
1936 \global\MT@inannottrue
1937 \MT@glet\MT@pdf@annot\@empty
1938 \MT@addto@annot{(line \number\inputlineno)}%
1939 \enddebug
1940 \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.

```
1941 \ifx\MT@font\@empty
1942 \let\MT@font\font@name
1943 \else
1944 \ifx\MT@font\font@name \else
1945 \debug \MT@addto@annot{= substituted with \MT@font}%
1946 \MT@register@subst@font
1947 \fi
1948 \fi
1949 \MT@setupfont
1950 \endgroup
1951 \endpackage
1952 }
1953 \*package
```

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

```
1954 \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, \LaTeX takes care that the fonts used for the `\accent` are already set up, so that we cannot be overlooking them. At first, I was going to change `\hmode@bgroup` only, but that is also used in the commands defined by `\DeclareTextFontCommand`, i. e., `\textit` etc.

```
1955 \let\MT@orig@add@accent\add@accent
1956 \def\add@accent#1#2{%
1957 \let\pickup@font\MT@orig@pickupfont
1958 \MT@orig@add@accent{#1}{#2}%
1959 \let\pickup@font\MT@pickupfont
1960 }
1961 \endpackage
1962 }
1963 \*package
```

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

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

```
1964 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

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

```
1965 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
```

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

```
1966 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}
```

14.2.10 Context-sensitive Setup

Here are the variants for context-sensitive setup.

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

```
1967 \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.

```
1968 \def\MT@check@font@cx{%
1969   \MT@if@true
1970   \MT@map@clist@c\MT@active@features{%
1971     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
1972     \MT@font \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1973     \ifMT@inlist@
1974       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
1975     \else
1976       \MT@if@false
1977     \fi
1978   }%
1979   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
1980 }
```

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

```
1981 \def\MT@register@subst@font@cx{%
1982   \MT@map@clist@c\MT@active@features{%
1983     \expandafter\MT@xadd
1984     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1985     {\font@name,}%
1986   }%
1987 }
```

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

```
1988 \def\MT@register@font@cx{%
1989   \MT@map@clist@c\MT@active@features{%
1990     \expandafter\ifx\csname MT@\@nameuse{MT@abbr@##1}\endcsname\relax\else
1991     \expandafter\MT@xadd
1992     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
1993     {\MT@font,}%
1994     \def\@tempa{##1}%
1995     \expandafter\MT@map@tlist@c
1996     \csname MT@##1@doc@contexts\endcsname
1997     \MT@rem@from@lists
1998   \fi
1999   }%
2000 }
```

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

```
2001 \def\MT@rem@from@lists#1{%
2002   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
2003     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
```

```

2004      \MT@font \csname MT@\@tempa @#1font@list\endcsname
2005    }%
2006 }

```

`\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.

```

2007 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}
2008 \MT@addto@setup{%
2009   \def\microtypecontext#1{%
2010     \MT@setup@contexts
2011     \let\MT@reset@context\relax
2012     \setkeys{MTC}{#1}%
2013     \selectfont
2014     \MT@reset@context
2015   }%
2016 }

```

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

```

\MT@reset@context@
2017 \def\MT@reset@context@{%
2018   \MT@vinfo{<<< Resetting contexts\on@line
2019   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
2020           /\MT@tr@context/\MT@kn@context/\MT@sp@context
2021   }%
2022   \selectfont
2023 }

```

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

```

2024 \def\MT@setup@contexts{%
2025   \MT@map@clic@MT@active@features
2026   {\global\MT@let@nc{MT@#1@font@list}\MT@font@list}%
2027   \MT@gl@et\MT@check@font\MT@check@font@cx
2028   \MT@gl@et\MT@register@font\MT@register@font@cx
2029   \MT@gl@et\MT@register@subst@font\MT@register@subst@font@cx
2030   \MT@gl@et\MT@setup@contexts\relax
2031 }

```

`\MT@define@context`

```

2032 \def\MT@define@context#1{%
2033   \define@key{MTC}{#1}[] {%
2034     \KV@sp@def\@tempb{#1}%
2035     \edef\@tempb{\@nameuse{MT@rbba@\@tempb}}%
2036     \MT@exp@one@n\MT@in@clic@\@tempb\MT@active@features
2037     \ifMT@inlist@

```

Using an empty context is only asking for trouble, therefore we choose the ‘@’ instead (hoping for the L^AT_EX users’ natural awe of this character).

```

2038   \MT@ifempty{##1}{\def\MT@val{@}}{\KV@sp@def\MT@val{##1}}%
2039   \expandafter\ifx\csname MT@\@tempb @context\endcsname\MT@val
2040 <debug> \MT@dinfo{1}{>>> no change of #1 context: `~\MT@val'}%
2041   \else
2042     \MT@vinfo{>>> Changing #1 context to `~\MT@val'\MessageBreak\on@line
2043 <debug>     \space(previous: `~\@nameuse{MT@\@tempb @context}')}%
2044     }%
2045     \def\MT@reset@context{\aftergroup\MT@reset@context@}%

```

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

```

2046   \global\MT@let@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes}%

```

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

```

2047 \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
2048 \MT@val \csname MT@\@tempb @doc@contexts\endcsname
2049 \ifMT@inlist@ \else
2050 \expandafter\MT@xadd\csname MT@\@tempb @doc@contexts\endcsname{{\MT@val}}%
2051 (debug) \MT@edinfo{1}{||| added #1 context: \nameuse{MT@\@tempb @doc@contexts}}%
2052 \fi
2053 \MT@edef@n{MT@\@tempb @context}{\MT@val}%
2054 \fi
2055 \fi
2056 }%
2057 }
2058 \MT@map@clist@c\MT@features@long{\MT@define@context{#1}}

```

\MT@pr@context Initialise the contexts.

```

\MT@ex@context 2059 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
\MT@tr@context 2060 \MT@def@n{MT@#1@context}{@}%
\MT@sp@context 2061 \MT@def@n{MT@#1@doc@contexts}{@}%
2062 }
\MT@kn@context 2063 \let\MT@extra@context\@empty

```

14.3 Configuration

14.3.1 Font Sets

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

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

```

2064 \def\DeclareMicrotypeSet{%
2065 \ifstar
2066 {\ifnextchar[\MT@DeclareSetAndUseIt
2067 {\MT@DeclareSetAndUseIt[]}}%
2068 {\ifnextchar[\MT@DeclareSet
2069 {\MT@DeclareSet[]}}%
2070 }

```

\MT@DeclareSet

```

\MT@DeclareSetAndUseIt 2071 \def\MT@DeclareSet[#1]{%
2072 \MT@DeclareSet@{#1}%
2073 }
2074 \def\MT@DeclareSetAndUseIt[#1]#2#3{%
2075 \MT@DeclareSet@{#1}{#2}{#3}%
2076 \UseMicrotypeSet[#1]{#2}%
2077 }

```

\MT@DeclareSet@

```

2078 \def\MT@DeclareSet@#1#2#3{%
2079 \KV@sp@def\@tempa{#1}%
2080 \MT@ifempty\@tempa{%
2081 \MT@map@clist@c\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%
2082 }%
2083 \MT@map@clist@c\@tempa{%
2084 \KV@sp@def\@tempa{##1}%
2085 \MT@ifempty\@tempa\relax{%
2086 \MT@is@feature{set declaration `#2'}%

```

```

2087      \MT@exp@one@n\MT@declare@sets
2088      {\csname MT@rbba@tempa\endcsname}{#2}{#3}%
2089    }%
2090  }%
2091 }%
2092 }%
2093 }

```

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

```
2094 \let\MT@curr@set@name\empty
```

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

```

2095 \def\MT@declare@sets#1#2#3{%
2096   \KV@sp@def\MT@curr@set@name{#2}%
2097   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
2098     \MT@warning{Redefining \nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2099     \global\MT@let@nc{MT@#1@list@size@\MT@curr@set@name}\empty
2100   }%
2101   \global\MT@let@nc{MT@#1@set@@\MT@curr@set@name}\empty
2102   <debug>\MT@dinfo{1}{declaring \nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2103   \setkeys{MT@#1@set}{#3}%
2104 }

```

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

```

2105 \def\MT@define@set@keys#1{%
2106   \MT@define@set@key@{encoding}{#1}%
2107   \MT@define@set@key@{family}{#1}%
2108   \MT@define@set@key@{series}{#1}%
2109   \MT@define@set@key@{shape}{#1}%
2110   \MT@define@set@key@size{#1}%
2111   \MT@define@set@key@font{#1}%
2112 }

```

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

```

2113 \def\MT@define@set@key@#1#2{%
2114   \define@key{MT@#2@set}{#1}[]{%
2115     \global\MT@let@nc{MT@#2@list@#1@\MT@curr@set@name}\empty
2116     \MT@map@clist@n{#1}{%
2117       \KV@sp@def\MT@val{####1}%
2118       \MT@get@highlevel{#1}%

```

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

```

2119   \MT@exp@two@n@g@addto@macro
2120   {\csname MT@#2@list@#1@\MT@curr@set@name\expandafter\endcsname}%
2121   {\MT@val,%}%
2122 }%

```

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

```

2123   \expandafter@g@addto@macro\expandafter\MT@font@sets
2124   {\csname MT@#2@list@#1@\MT@curr@set@name\endcsname
2125   <debug>\MT@dinfo{n1}{1}{-- #1: \nameuse{MT@#2@list@#1@\MT@curr@set@name}}}%
2126   }%
2127 }

```

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

```

2128 \def\MT@get@highlevel#1{%
2129   \expandafter\MT@test@ast\MT@val*\@nil{%

```

And ‘family = *’ will become \familydefault.

```

2130   \MT@ifempty\tempa{\def\tempa{#1}}\relax
2131   \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.

```

2132 }%
2133 }

\MT@test@ast Test whether last character is an asterisk.
2134 \def\MT@test@ast#1*#2\@nil{%
2135   \def\@tempa{#1}%
2136   \MT@ifempty{#2}\@gobble\@firstofone
2137 }

\MT@font@sets Fully expand the font specification and fix catcodes for all font sets.
\MT@fix@font@set 2138 \let\MT@font@sets\@empty
2139 \def\MT@fix@font@set#1{%
2140   \xdef#1{#1}%
2141   \global\MT@make@string#1%
2142 }

\MT@define@set@key@size size requires special treatment.
2143 \def\MT@define@set@key@size#1{%
2144   \define@key{MT@#1@set}{size}[]{%
2145     \MT@map@clist@n{##1}{%
2146       \KV@sp@def\MT@val{###1}%
2147       \expandafter\MT@get@range\MT@val--\@nil
2148       \ifx\MT@val\relax \else
2149         \expandafter\MT@xadd
2150         \csname MT@#1list@size@\MT@curr@set@name\endcsname
2151         {{{\MT@lower}{\MT@upper}\relax}}%
2152       \fi
2153     }%
2154     \debug\MT@dinfo@n1{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
2155   }%
2156 }
```

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

```

\MT@get@range Ranges will be stored as triplets of {<lower bound>}{<upper bound>}{<list name>}.
\MT@upper For simple sizes, the upper boundary is -1.
\MT@lower 2157 \def\MT@get@range#1-#2-#3\@nil{%
2158   \MT@ifempty{#1}{%
2159     \MT@ifempty{#2}{%
2160       \let\MT@val\relax
2161     }%
2162     \def\MT@lower{0}%
2163     \def\MT@val{#2}%
2164     \MT@get@size
2165     \edef\MT@upper{\MT@val}%
2166   }%
2167 }%
2168 \def\MT@val{#1}%
2169 \MT@get@size
2170 \ifx\MT@val\relax \else
2171   \edef\MT@lower{\MT@val}%
2172   \MT@ifempty{#2}{%
2173     \MT@ifempty{#3}%
2174     {\def\MT@upper{-1}}%

```

2048pt is TeX's maximum font size.

```

2175     {\def\MT@upper{2048}}%
2176     }{%
2177     \def\MT@val{#2}%
2178     \MT@get@size
2179     \ifx\MT@val\relax \else
2180       \MT@ifdim\MT@lower>\MT@val{%
2181         \MT@warning{%
2182           Invalid size range (\MT@lower\space > \MT@val) in font set
2183           ~\MT@curr@set@name'.\MessageBreak Swapping sizes}%
2184         \edef\MT@upper{\MT@lower}%
2185         \edef\MT@lower{\MT@val}%
2186       }{%
2187         \edef\MT@upper{\MT@val}%
2188       }%
2189       \MT@ifdim\MT@lower=\MT@upper
2190       {\def\MT@upper{-1}}%
2191       \relax
2192     \fi
2193   }%
2194 \fi
2195 }%
2196 }
```

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

```
2197 \def\MT@get@size{%
```

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

```

2198 \if*\MT@val\relax
2199   \def\@tempa{\normalsize}%
2200 \else
2201   \MT@let@cn\@tempa{\MT@val}%
2202 \fi
2203 \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).

```

2204 \begingroup
2205   \def\set@fontsize#1#2#3#4\@nil{\gdef\MT@val{##2}}%
2206   \@tempa\@nil
2207 \endgroup
2208 \fi
```

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

```

2209 \MT@ifdimen\MT@val{%
2210   \@defaultunits\@tempdima\MT@val pt\relax\@nni
2211   \edef\MT@val{\strip@pt\@tempdima}%
2212 }{%
2213   \MT@warning{Could not parse font size ~\MT@val'\MessageBreak
2214             in font set ~\MT@curr@set@name'}%
2215   \let\MT@val\relax
2216 }%
2217 }
```

\MT@define@set@key@font

```
2218 \def\MT@define@set@key@font#1{%
```

```

2219 \define@key{MT@#1@set}{font}[]{%
2220   \global\MT@let@enc{MT@#1list@font@MT@curr@set@name}\@empty
2221   \MT@map@clist@n{#1}{%
2222     \KV@esp@def\MT@val{###1}%
2223     \expandafter\MT@get@font\MT@val///// \@nil
2224     \MT@exp@two@n@g@addto@macro
2225     {\csname MT@#1list@font@MT@curr@set@name\expandafter\endcsname}%
2226     {\MT@val},}%
2227   }%
2228   \expandafter\g@addto@macro\expandafter\MT@font@sets
2229   \csname MT@#1list@font@MT@curr@set@name\endcsname
2230 (debug)\MT@info@n1{1}{-- font: \@nameuse{MT@#1list@font@MT@curr@set@name}}%
2231 }%
2232 }

```

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

```

2233 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2234   \MT@ifempty{#1#2#3#4#5}\relax{%
2235     \let\@tempb\@empty
2236     \def\MT@temp{#1/#2/#3/#4/#5}%
2237     \MT@get@axis{encoding}{#1}%
2238     \MT@get@axis{family}{#2}%
2239     \MT@get@axis{series}{#3}%
2240     \MT@get@axis{shape}{#4}%
2241     \MT@ifempty{#5}{%
2242       \MT@warn@axis@empty{size}{\string\normalsize}%
2243       \def\MT@val{*}%
2244     }{%
2245       \def\MT@val{#5}%
2246     }%
2247     \MT@get@size
2248     \ifx\MT@val\relax\def\MT@val{0}\fi
2249     \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2250     \let\MT@val\@tempb
2251   }%
2252 }

```

`\MT@get@axis`

```

2253 \def\MT@get@axis#1#2{%
2254   \def\MT@val{#2}%
2255   \MT@get@highlevel{#1}%
2256   \MT@ifempty\MT@val{%
2257     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2258     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2259   }\relax
2260   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2261 }

```

`\MT@warn@axis@empty`

```

2262 \def\MT@warn@axis@empty#1#2{%
2263   \MT@warning{#1 axis is empty in font specification\MessageBreak
2264     ~\MT@temp'. Using ~#2' instead}%
2265 }

```

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

```
2266 \MT@map@clist@c\MT@features{\MT@define@set@keys{#1}}
```

It is also used for `\DisableLigatures`.

```
2267 \MT@define@set@keys{n1}
```

`\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.

```

2268 \renewcommand*\UseMicrotypeSet[2][]{%
2269   \KV@sp@def\@tempa{#1}%
2270   \MT@ifempty\@tempa{%
2271     \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
2272   }{%
2273     \MT@map@clist@c\@tempa{%
2274       \KV@sp@def\@tempa{##1}%
2275       \MT@ifempty\@tempa\relax{%
2276         \MT@is@feature{activation of set `#2'}{%
2277           \MT@exp@one@n\MT@use@set
2278             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2279         }%
2280       }%
2281     }%
2282   }%
2283 }

```

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

```

\MT@ex@setname 2284 \def\MT@use@set#1#2{%
\MT@tr@setname 2285   \KV@sp@def\@tempa{#2}%
2286   \MT@ifdefined@n@TF{MT@#1@set@{\@tempa}}{%
\MT@sp@setname 2287     \MT@xdef@n{MT@#1@setname}{\@tempa}%
\MT@kn@setname 2288     \MT@info{Using \@nameuse{MT@abbr@#1} set \@tempa'}%
2289   }{%
2290     \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2291       \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2292     }%
2293     \MT@warning{%
2294       The \@nameuse{MT@abbr@#1} set \@tempa' is undeclared.\MessageBreak
2295       Using set \@nameuse{MT@#1@setname}' instead}%
2296   }%
2297 }

```

`\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'.

```

2298 \MT@requires@pdftex5{
2299   \renewcommand*\DisableLigatures[1]{%
2300     \edef\MT@active@features{\MT@active@features,nl}%
2301     \MT@noligaturestrue
2302     \MT@declare@sets{nl}{no ligatures}{#1}%
2303     \gdef\MT@nl@setname{no ligatures}%
2304   }
2305 }{

```

If pdf_{TEX} is too old, we issue a warning and neutralise the command.

```

2306 \renewcommand*\DisableLigatures[1]{%
2307   \MT@warning{Disabling ligatures of a font is only possible\MessageBreak
2308     with pdftex version 1.30 or newer.\MessageBreak
2309     Ignoring \string\DisableLigatures}%
2310   \MT@glet\DisableLigatures\@gobble
2311 }
2312 }

```

`\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.

```

2313 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2314   \KV@sp@def\@tempa{#1}%
2315   \MT@ifempty\@tempa{%
2316     \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
2317   }{%
2318     \MT@map@clist@c\@tempa{%
2319       \KV@sp@def\@tempa{##1}%

```

```

2320      \MT@ifempty\@tempa\relax{%
2321      \MT@is@feature{declaration of default set `#2'}{%
2322      \MT@exp@one@n\MT@set@default@set
2323      {\csname MT@rbba@\@tempa\endcsname}{#2}%
2324      }%
2325      }%
2326      }%
2327      }%
2328      }

\MT@default@pr@set
\MT@default@ex@set 2329 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 2330 \KV@sp@def\@tempa{#2}%
\MT@default@sp@set 2331 \MT@ifdefined@n@TF{MT@#1@set@\@tempa}{%
\MT@default@kn@set 2332 (debug)\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set \@tempa}%
\MT@set@default@set 2333 \MT@xdef@n{MT@default@#1@set}{\@tempa}%
2334 }{%
2335 \MT@warning{%
2336 The \@nameuse{MT@abbr@#1} set \@tempa' is not declared.\MessageBreak
2337 Cannot make it the default set. Using set\MessageBreak `all' instead}%
2338 \MT@xdef@n{MT@default@#1@set}{all}%
2339 }%
2340 }

```

\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.

```

2341 \renewcommand*\DeclareMicrotypeAlias[2]{%
2342 \KV@sp@def\@tempa{#1}%
2343 \KV@sp@def\@tempb{#2}%
2344 \MT@make@string\@tempb
2345 \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2346 \MT@warning{Alias font family \@tempb' will override
2347 alias \@nameuse{MT@\@tempa @alias}'\MessageBreak
2348 for font family \@tempa'}}%
2349 \MT@xdef@n{MT@\@tempa @alias}{\@tempb}%

```

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

```

2350 \MT@ifdefined@c@T\MT@family{%
2351 (debug)\MT@dinfo{1}{Activating alias font \@tempb' for \@tempa'}%
2352 \MT@glet\MT@familyalias\@tempb
2353 }%
2354 }

```

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

```

2355 \def\LoadMicrotypeFile#1{%
2356 \KV@sp@def\@tempa{#1}%
2357 \MT@make@string\@tempa
2358 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2359 \ifMT@inlist@
2360 \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2361 \else
2362 \MT@xadd\MT@file@list{\@tempa,}%
2363 \MT@begin@catcodes
2364 \InputIfFileExists{mt-\@tempa.cfg}{%
2365 \edef\MT@curr@file{mt-\@tempa.cfg}%
2366 \MT@vinfo{... Loading configuration file \MT@curr@file}%
2367 }{%
2368 \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2369 does not exist}%

```

```

2370 }%
2371 \MT@end@catcodes
2372 \fi
2373 }

```

14.3.2 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.

```

2374 \def\DeclareMicrotypeBabelHook#1#2{%
2375   \MT@map@clist@n{#1}{%
2376     \KV@sp@def\@tempa{#1}%
2377     \MT@gdef@n{MT@babel@\@tempa}{#2}%
2378   }%
2379 }

```

14.3.3 Fine Tuning

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

`\SetProtrusion` This macro accepts three arguments: [options,] set of font attributes and list of character protrusion factors.

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

```

2380 \def\SetProtrusion{%
2381   \MT@begin@catcodes
2382   \MT@SetProtrusion
2383 }

```

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

```

\MT@pr@c@name 2384 \newcommand*\MT@SetProtrusion[3][]{%

```

```

\MT@extra@context 2385 \let\MT@extra@context\empty

```

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

```

2386 \MT@set@named@keys{MT@pr@c}{#1}%
2387 <debug> \MT@dinfo{1}{creating protrusion list `\'MT@pr@c@name'}%
2388 \def\MT@permute@list{pr@c}%
2389 \setkeys{MT@cfg}{#2}%

```

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

```

2390 \MT@permute

```

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

```

2391 \MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}%
2392 \MT@end@catcodes
2393 }

```

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

```

2394 \def\SetExpansion{%
2395   \MT@begin@catcodes
2396   \MT@SetExpansion
2397 }

```

```

\MT@SetExpansion
  \MT@ex@c@name 2398 \newcommand*\MT@SetExpansion[3] [] {%
\MT@extra@context 2399 \let\MT@extra@context\@empty
  \MT@permutelist 2400 \MT@set@named@keys{MT@ex@c}{#1}%
2401 \MT@ifdefined@n{T{MT@ex@c@MT@ex@c@name @factor}}{%
2402 \ifnum\csname MT@ex@c@MT@ex@c@name @factor\endcsname > \@m
2403 \MT@warning@n1{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
2404 too large in list\MessageBreak ``\MT@ex@c@name'. Setting it to the
2405 maximum of 1000}%
2406 \global\MT@let@nc{MT@ex@c@MT@ex@c@name @factor}\@m
2407 \fi
2408 }%
2409 <debug>\MT@dinfo{1}{creating expansion list ``\MT@ex@c@name'}%
2410 \def\MT@permutelist{ex@c}%
2411 \setkeys{MT@cfg}{#2}%
2412 \MT@permute
2413 \MT@gdef@n{MT@ex@c@MT@ex@c@name}{#3}%
2414 \MT@end@catcodes
2415 }

\SetTracking
2416 \def\SetTracking{%
2417 \MT@begin@catcodes
2418 \MT@SetTracking
2419 }

\MT@SetTracking
2420 \newcommand*\MT@SetTracking[3] [] {%
2421 \let\MT@extra@context\@empty
2422 \MT@set@named@keys{MT@tr@c}{#1}%
2423 <debug>\MT@dinfo{1}{creating tracking list ``\MT@tr@c@name'}%
2424 \def\MT@permutelist{tr@c}%
2425 \setkeys{MT@cfg}{#2}%
2426 \MT@permute
2427 \MT@gdef@n{MT@tr@c@MT@tr@c@name}{#3}%
2428 \MT@end@catcodes
2429 }

\SetExtraSpacing
2430 \def\SetExtraSpacing{%
2431 \MT@begin@catcodes
2432 \MT@SetExtraSpacing
2433 }

\MT@SetExtraSpacing
  \MT@sp@c@name 2434 \newcommand*\MT@SetExtraSpacing[3] [] {%
\MT@extra@context 2435 \let\MT@extra@context\@empty
  \MT@permutelist 2436 \MT@set@named@keys{MT@sp@c}{#1}%
2437 <debug>\MT@dinfo{1}{creating spacing list ``\MT@sp@c@name'}%
2438 \def\MT@permutelist{sp@c}%
2439 \setkeys{MT@cfg}{#2}%
2440 \MT@permute
2441 \MT@gdef@n{MT@sp@c@MT@sp@c@name}{#3}%
2442 \MT@end@catcodes
2443 }

\SetExtraKerning
2444 \def\SetExtraKerning{%
2445 \MT@begin@catcodes
2446 \MT@SetExtraKerning
2447 }

```

```

\MT@SetExtraKerning
  \MT@kn@c@name 2448 \newcommand*\MT@SetExtraKerning[3] [] {%
\MT@extra@context 2449 \let\MT@extra@context\@empty
  \MT@permutelist 2450 \MT@set@named@keys{MT@kn@c}{#1}%
2451 (debug)\MT@dinfo{1}{creating kerning list ~\MT@kn@c@name'}%
2452 \def\MT@permutelist{kn@c}%
2453 \setkeys{MT@c@fg}{#2}%
2454 \MT@permute
2455 \MT@gdef@n{MT@kn@c\MT@kn@c@name}{#3}%
2456 \MT@end@catcodes
2457 }

\MT@set@named@keys We first set the name (if specified), then remove it from the list, and set the
\MT@options remaining keys.
2458 \def\MT@set@named@keys#1#2{%
2459 \def\x##1name=##2,##3\@nil{%
2460 \setkeys{#1}{name=##2}%
2461 \def\MT@options{##1##3}%
2462 \MT@rem@from@clist{name=}\MT@options
2463 }%
2464 \x#2,name=,\@nil
2465 \expandtwoargs\setkeys{#1}\MT@options
2466 }

\MT@define@code@key Define the keys for the configuration lists (which are setting the codes, in pdfTeX
speak).
2467 \def\MT@define@code@key#1#2{%
2468 \define@key{MT@#2}{#1} [] {%
2469 \@tempcnta=\@ne
2470 \MT@map@clist@n{##1}{%
2471 \KV@sp@def\MT@val{###1}%
2472 \MT@get@highlevel{#1}%
2473 \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
2474 \advance\@tempcnta \@ne
2475 }%
2476 }%
2477 }

\MT@define@code@key@size \MT@tempsize must be in a \csize, so that it is at least \relax, not undefined.
2478 \def\MT@define@code@key@size#1{%
2479 \define@key{MT@#1}{size} [] {%
2480 \MT@map@clist@n{##1}{%
2481 \KV@sp@def\MT@val{###1}%
2482 \expandafter\MT@get@range\MT@val--\@nil
2483 \ifx\MT@val\relax \else
2484 \expandafter\MT@xadd\csize MT@tempsize\endcsize
2485 {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2486 \fi
2487 }%
2488 }%
2489 }

\MT@define@code@key@font
2490 \def\MT@define@code@key@font#1{%
2491 \define@key{MT@#1}{font} [] {%
2492 \MT@map@clist@n{##1}{%
2493 \KV@sp@def\MT@val{###1}%
2494 \expandafter\MT@get@font@and@size\MT@val///// \@nil
2495 \MT@xdef@n{MT@\MT@permutelist @\@tempb\MT@extra@context}%

```



```

2496         {\csname MT@MT@permutelist @name\endcsname}%
2497 (*debug)
2498         \MT@info@n1{1}{initialising: use list for font \@tempb=\MT@val
2499                 \ifx\MT@extra@context\@empty\else\MessageBreak
2500                 (context: \MT@extra@context)\fi}%
2501 (/debug)
2502         \expandafter\MT@xaddb
2503         \csname MT@MT@permutelist @\@tempb\MT@extra@context @sizes\endcsname
2504         {{{\MT@val}\m@ne}\MT@curr@set@name}}}%
2505     }%
2506 }%
2507 }

```

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

```

2508 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2509     \MT@ifempty{#1#2#3#4#5}\relax{%
2510         \let\@tempb\@empty
2511         \def\MT@temp{#1/#2/#3/#4/#5}%
2512         \MT@get@axis{encoding}{#1}%
2513         \MT@get@axis{family}{#2}%
2514         \MT@get@axis{series}{#3}%
2515         \MT@get@axis{shape}{#4}%

```

Append an asterisk for the size.

```

2516         \edef\@tempb{\@tempb*}%
2517         \MT@ifempty{#5}{%
2518             \MT@warn@axis@empty{size}{\string\normalsize}%
2519             \def\MT@val{*}%
2520         }{%
2521             \def\MT@val{#5}%
2522         }%
2523         \MT@get@size
2524     }%
2525 }

2526 \MT@define@code@key{encoding}{cfg}
2527 \MT@define@code@key{family}{cfg}
2528 \MT@define@code@key{series}{cfg}
2529 \MT@define@code@key{shape}{cfg}
2530 \MT@define@code@key@size{cfg}
2531 \MT@define@code@key@font{cfg}

```

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

```

2532 \def\MT@define@opt@keys#1{%

```

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

```

2533     \define@key{MT@#1c}{name}[]{%
2534         \MT@ifempty{##1}{%
2535             \MT@edef\MT@#1c@name{\MT@curr@file/\the\inputlineno}%
2536         }{%
2537             \MT@edef\MT@#1c@name{##1}%
2538             \MT@ifdefined@nT{MT@#1c@csname MT@#1c@name\endcsname}{%
2539                 \MT@warning{Redefining \@nameuse{MT@abbr@#1} list ~\@nameuse{MT@#1c@name}}}%
2540             }%
2541         }%
2542         \MT@let@cn\MT@curr@set@name{MT@#1c@name}%
2543     }%
2544     \MT@define@opt@key{#1}{load}%
2545     \MT@define@opt@key{#1}{factor}%
2546     \MT@define@opt@key{#1}{preset}%
2547     \MT@define@opt@key{#1}{inputenc}%

```

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

```
2548 \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
2549 }
```

\MT@define@opt@key

```
2550 \def\MT@define@opt@key#1#2{%
2551 \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
2552 \MT@xdefon{MT@#1@c@MT@curr@set@name @#2}{##1}}}%
2553 }
2554 \MT@map@clist@c\MT@features{\MT@define@opt@keys{#1}}
```

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

```
2555 \define@key{MT@pr@c}{unit}[character]{%
2556 \global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
2557 \def\@tempa{#1}%
2558 \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.

```
2559 \MT@ifdimen\@tempa{%
2560 \global\MT@let@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa
2561 }{%
2562 \MT@warning{'\@tempa' is not a dimension.\MessageBreak
2563 Ignoring it and setting values relative to\MessageBreak
2564 character widths}%
2565 }%
2566 }%
2567 }
```

Tracking may only be relative to a dimension.

```
2568 \define@key{MT@tr@c}{unit}[1em]{%
2569 \global\MT@let@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
2570 \def\@tempa{#1}%
2571 \MT@ifdimen\@tempa{%
2572 \global\MT@let@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa
2573 }{%
2574 \MT@warning{'\@tempa' is not a dimension.\MessageBreak
2575 Ignoring it and setting values relative to\MessageBreak
2576 1em}%
2577 \MT@gdefon{MT@tr@c@MT@curr@set@name @unit}{1em}%
2578 }%
2579 }
```

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

```
2580 \def\MT@define@key@unit#1{%
2581 \define@key{MT@#1@c}{unit}[space]{%
2582 \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
2583 \def\@tempa{##1}%
2584 \MT@ifstreq\@tempa{character}\relax{%
2585 \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
2586 \MT@ifstreq\@tempa{space}\relax{%
2587 \MT@ifdimen\@tempa{%
2588 \global\MT@let@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa
2589 }{%
2590 \MT@warning{'\@tempa' is not a dimension.\MessageBreak
2591 Ignoring it and setting values relative to\MessageBreak
2592 width of space}%
2593 }%
2594 }%
2595 }%
2596 }
```

```

2597 }
2598 \MT@define@key@unit{sp}
2599 \MT@define@key@unit{kn}

```

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

```

2600 \def\MT@define@ex@opt@key#1{%
2601   \define@key{MT@ex@c}{#1}[]{%
2602     \MT@ifempty{##1}\relax{%
2603       \MT@ifint{##1}%

```

A space terminates the number.

```

2604     \MT@gdef@n{MT@ex@c@MT@curr@set@name @#1}{##1 }%
2605   }{%
2606     \MT@warning{%
2607       Value `##1' for option `#1' is not a number.\MessageBreak
2608       Ignoring it}%
2609   }%
2610 }%
2611 }%
2612 }

```

```

2613 \MT@define@ex@opt@key{stretch}
2614 \MT@define@ex@opt@key{shrink}
2615 \MT@define@ex@opt@key{step}
2616 \define@key{MT@ex@c}{auto}[true]{%
2617   \def\@tempa{#1}%
2618   \csname if\@tempa\endcsname

```

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

```

2619   \MT@requires@pdftex4{%
2620     \MT@gdef@n{MT@ex@c@MT@curr@set@name @auto}{autoexpand}%
2621   }{%
2622     \MT@warning{pdftex too old for automatic font expansion}%
2623   }
2624 \else
2625   \MT@requires@pdftex4{%
2626     \global\MT@let@nc{MT@ex@c@MT@curr@set@name @auto}\@empty
2627   }\relax
2628 \fi
2629 }

```

14.3.4 Character Inheritance

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

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

```

2630 \renewcommand*\DeclareCharacterInheritance[1][]{%
2631   \let\MT@extra@context\@empty
2632   \let\MT@extra@inputenc\@undefined
2633   \let\MT@inh@feat\@empty
2634   \setkeys{MT@inh@}{#1}%
2635   \MT@begin@catcodes
2636   \MT@set@inh@list
2637 }

```

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

```

2638 \def\MT@set@inh@list#1#2{%
2639   \MT@ifempty\MT@inh@feat{%
2640     \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{##1}{##2}}}%
2641   }%
2642   \MT@map@clist@c\MT@inh@feat{%
2643     \KV@esp@def\@tempa{##1}%
2644     \MT@ifempty\@tempa\relax{%
2645       \MT@exp@one@n\MT@declare@char@inh
2646       {\csname MT@rbba@\@tempa\endcsname}{##1}{##2}%
2647     }%
2648   }%
2649 }%
2650 \MT@end@catcodes
2651 }
```

The keys for the optional argument.

```

2652 \MT@map@clist@c\MT@features@long{%
2653   \define@key\MT@inh@{##1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}%
2654   \define@key\MT@inh@{inputenc}{\def\MT@extra@inputenc{##1}}
```

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

```

2655 \def\MT@declare@char@inh#1#2#3{%
2656   \MT@edef@n{\MT@#1@inh@name}%
2657   {\MT@curr@file/\the\inputlineno (\@nameuse{\MT@abbr@#1})}%
2658   \MT@let@cn\MT@curr@set@name{\MT@#1@inh@name}%
2659   \MT@ifdefined@c@T\MT@extra@inputenc{%
2660     \MT@xdef@n{\MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
2661   (debug) \MT@dinfo{1}{creating inheritance list ` \@nameuse{\MT@#1@inh@name}'}%
2662   \MT@gdef@n{\MT@#1@inh@\csname MT@#1@inh@name\endcsname}{##3}%
2663   \def\MT@permute@list{##1@inh}%
2664   \setkeys{\MT@inh}{##2}%
2665   \MT@permute
2666 }
```

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

```

2667 \define@key{\MT@inh}{encoding}[]{%
2668   \def\MT@val{##1}%
2669   \expandafter\MT@encoding@check\MT@val,\@nil
2670   \MT@get@highlevel{encoding}%
2671   \MT@edef@n{\MT@tempencoding1}{\MT@val}%
2672 }
```

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

```

2673 \def\MT@encoding@check#1,#2\@nil{%
2674   \MT@ifempty{##2}\relax{%
2675     \edef\MT@val{##1}%
2676     \MT@warning{You may only specify one encoding for character\MessageBreak
2677               inheritance lists. Ignoring encoding(s) ##2}%
2678   }%
2679 }
```

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

```

2680 \MT@define@code@key{family}{inh}
2681 \MT@define@code@key{series}{inh}
2682 \MT@define@code@key{shape}{inh}
2683 \MT@define@code@key{size}{inh}
2684 \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`).

```
2685 \def\MT@inh@do#1,{%
2686   \ifx\relax#1\@empty \else
2687     \MT@inh@split #1=\relax
2688     \expandafter\MT@inh@do
2689   \fi
2690 }
```

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

```
2691 \def\MT@inh@split#1=#2=#3\relax{%
2692   \def\@tempa{#1}%
2693   \ifx\@tempa\@empty \else
2694     \MT@get@slot
2695     \ifnum\MT@char > \m@ne
2696       \let\MT@val\MT@char
2697       \MT@map@clist@n{#2}{%
2698         \def\@tempa{#1}%
2699         \ifx\@tempa\@empty \else
2700           \MT@get@slot
2701           \ifnum\MT@char > \m@ne
2702             \expandafter\MT@xadd
2703             \csname MT@inh@MT@listname @\MT@val @\endcsname
2704             {\MT@char}%
2705           \fi
2706         \fi
2707       }%
2708   \fi
2709   \MT@dinfo@n{2}{children of #1 (\MT@val):
2710     \@nameuse{MT@inh@MT@listname @\MT@val @}}%
2711   \fi
2712   \fi
2713   \fi
2714 }
```

14.3.5 Permutation

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

```
2715 \def\MT@permute{%
2716   \let\MT@cnt@encoding\@ne
2717   \MT@permute@
```

Undefine commands for the next round.

```
2718   \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
2719   \MT@gllet\MT@temp@size\undefined
2720 }
2721 \def\MT@permute@{%
2722   \let\MT@cnt@family\@ne
2723   \MT@permute@@
2724   \MT@increment\MT@cnt@encoding
2725   \MT@ifdefined@n@T{MT@temp@encoding\MT@cnt@encoding}%
2726   \MT@permute@
2727 }
```

```

2728 \def\MT@permute@@{%
2729   \let\MT@cnt@series\@ne
2730   \MT@permute@@@
2731   \MT@increment\MT@cnt@family
2732   \MT@ifdefined@n@T\MT@tempfamily\MT@cnt@family}%
2733   \MT@permute@@
2734 }
2735 \def\MT@permute@@@{%
2736   \let\MT@cnt@shape\@ne
2737   \MT@permute@@@
2738   \MT@increment\MT@cnt@series
2739   \MT@ifdefined@n@T\MT@tempseries\MT@cnt@series}%
2740   \MT@permute@@@
2741 }
2742 \def\MT@permute@@@@{%
2743   \MT@permute@@@@
2744   \MT@increment\MT@cnt@shape
2745   \MT@ifdefined@n@T\MT@tempshape\MT@cnt@shape}%
2746   \MT@permute@@@@
2747 }

```

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

```

2748 \def\MT@permute@@@@@{%
2749   \MT@permute@define{encoding}%
2750   \ifMT@document
2751     \ifx\MT@tempencoding\@empty \else
2752       \MT@ifdefined@n@TF{T@MT@tempencoding}\relax
2753       {\expandafter\expandafter\expandafter\@gobble}%
2754     \fi
2755   \fi
2756   \MT@permute@@@@@
2757 }

```

\MT@permute@@@@@

```

2758 \def\MT@permute@@@@@@{%
2759   \MT@permute@define{family}%
2760   \MT@permute@define{series}%
2761   \MT@permute@define{shape}%
2762   \edef\@tempa{\MT@tempencoding
2763     /\MT@tempfamily
2764     /\MT@tempseries
2765     /\MT@tempshape
2766     /\MT@ifdefined@c@T\MT@tempsize *}

```

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

```

2767   \def\@tempb{////}%
2768   \ifx\@tempa\@tempb \else
2769     \ifx\MT@tempencoding\@empty
2770       \MT@warning{%
2771         You have to specify an encoding for\MessageBreak
2772         \@nameuse{MT@abbr@MT@permute!list} list
2773         \@nameuse{MT@MT@permute!list @name}'.\MessageBreak
2774         Ignoring it}%
2775     \else
2776       \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

2777   \MT@ifdefined@n@T{MT@MT@permute!list \@tempa\MT@extra@context @size}{%
2778     \MT@map@tlist@c
2779     \MT@tempsize

```

```

2780         \MT@check@rlist
2781     }%
2782     \expandafter\MT@xaddb
2783         \csname MT@\MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
2784         \MT@tempsize
2785 <(*debug)
2786     \MT@info@n1{1}{initialising: use list for font \@tempa, \MessageBreak
2787         sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
2788             @sizes\endcsname}%
2789 </debug>
2790 }{%

```

Only one list can apply to a given combination.

```

2791     \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context}{%
2792         \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
2793             \@nameuse{MT@\MT@permutelist @name}' will override list \MessageBreak
2794             \@nameuse{MT@\MT@permutelist @\@tempa\MT@extra@context}'
2795             for font \@tempa}%
2796     }%
2797 <(*debug)
2798     \MT@info@n1{1}{initialising: use list for font \@tempa
2799         \ifx\MT@extra@context\empty\else \MessageBreak
2800             (context: \MT@extra@context)\fi}%
2801 </debug>
2802 }%
2803     \MT@xdef@n{MT@\MT@permutelist @\@tempa\MT@extra@context}%
2804     {\csname MT@\MT@permutelist @name\endcsname}%
2805     \fi
2806     \fi
2807 }

```

\MT@permute@define Define the commands.

```

2808 \def\MT@permute@define#1{%
2809     \expandafter\@tempcnta=\csname MT@cnt@#1\endcsname\relax
2810     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
2811     {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
2812     {\MT@let@nc{MT@temp#1}\empty}%
2813 }

```

\MT@permute@reset Reset the commands.

```

2814 \def\MT@permute@reset#1{%
2815     \@tempcnta=\@ne
2816     \MT@loop
2817         \MT@let@nc{MT@temp#1\the\@tempcnta}\undefined
2818         \advance\@tempcnta\@ne
2819     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
2820         \iftrue
2821         \iffalse
2822     \MT@repeat
2823 }

```

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

```

2824 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

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

```

2825 \def\MT@check@rlist@#1#2#3{%
2826     \def\@tempb{#1}%
2827     \def\@tempc{#2}%
2828     \MT@iffalse
2829     \expandafter\MT@map@tlist@c
2830     \csname MT@\MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname

```

```

2831     \MT@check@range
2832 }

\MT@check@range ... recurse through the list of existing ranges.
2833 \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.
2834 \def\MT@check@range@#1#2#3{%
2835     \MT@ifdim{#2}=\m@ne{%
2836         \MT@ifdim\@tempc=\m@ne{%

            • Both items are simple sizes.

2837             \MT@ifdim\@tempb={#1}\MT@iftrue\relax
2838         }{%

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

2839             \MT@ifdim\@tempb>{#1}\relax{%
2840                 \MT@ifdim\@tempc>{#1}{%
2841                     \MT@iftrue
2842                     \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
2843                 }\relax
2844             }%
2845         }%
2846     }{%
2847         \MT@ifdim\@tempc=\m@ne{%

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

2848             \MT@ifdim\@tempb<{#2}{%
2849                 \MT@ifdim\@tempb<{#1}\relax\MT@iftrue
2850             }\relax
2851         }{%

            • Both items are ranges.

2852             \MT@ifdim\@tempb<{#2}{%
2853                 \MT@ifdim\@tempc>{#1}{%
2854                     \MT@iftrue
2855                     \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
2856                 }\relax
2857             }\relax
2858         }%
2859     }%
2860     \ifMT@if@
2861         \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
2862             ~\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
2863             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.

2864         \expandafter\MT@tlist@break
2865     \fi
2866 }

```

14.4 Package Options

14.4.1 Declaring the Options

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

\ifMT@opt@auto

\ifMT@opt@DVI


```

2867 \newif\ifMT@opt@expansion
2868 \newif\ifMT@opt@auto
2869 \newif\ifMT@opt@DVI

```

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

```

2870 \def\MT@define@option#1{%
2871   \define@key{MT}{#1}[true]{%
2872     \csname MT@opt@#1true\endcsname
2873     \MT@map@clist@n{##1}{%
2874       \KV@sp@def\MT@val{###1}%
2875       \MT@ifempty\MT@val\relax{%
2876         \csname MT@#1true\endcsname
2877         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2878         \MT@ifstreq\MT@val{true}\relax
2879         {%
2880           \MT@ifstreq\MT@val{false}{%
2881             \csname MT@#1false\endcsname
2882           }{%
2883             \MT@ifstreq\MT@val{compatibility}{%
2884               \MT@let@nc{MT@\@tempb @level}\@ne
2885             }{%
2886               \MT@ifstreq\MT@val{nocompatibility}{%
2887                 \MT@let@nc{MT@\@tempb @level}\tw@
2888               }{%

```

If everything failed, it should be a set name.

```

2889       \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
2890         \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
2891       }{%
2892         \MT@xdef@n{MT@\@tempb @setname}%
2893         {\@nameuse{MT@default@\@tempb @set}}%
2894         \MT@warning@n1{%
2895           The #1 set '\MT@val' is undeclared.\MessageBreak
2896           Using set '\@nameuse{MT@\@tempb @setname}' instead}%
2897       }%
2898     }%
2899   }%
2900 }%
2901 }%
2902 }%
2903 }%
2904 }%
2905 }

2906 \MT@define@option{protrusion}
2907 \MT@define@option{expansion}

```

activate is a shortcut for protrusion and expansion.

```

2908 \define@key{MT}{activate}[true]{%
2909   \setkeys{MT}{protrusion={#1}}%
2910   \setkeys{MT}{expansion={#1}}%
2911 }

```

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

```

2912 \def\MT@define@option@#1{%
2913   \define@key{MT}{#1}[true]{%
2914     \csname MT@opt@#1true\endcsname
2915     \MT@map@clist@n{##1}{%
2916       \KV@sp@def\MT@val{###1}%
2917       \MT@ifempty\MT@val\relax{%
2918         \csname MT@#1true\endcsname
2919         \edef\@tempb{\csname MT@rbba@#1\endcsname}%

```

```

2920 \MT@ifstreq\MT@val{true}\relax
2921 {%
2922 \MT@ifstreq\MT@val{false}{%
2923 \csname MT@#1false\endcsname
2924 }{%
2925 \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
2926 \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
2927 }{%
2928 \MT@xdef@n{MT@\@tempb @setname}%
2929 {\@nameuse{MT@default@\@tempb @set}}%
2930 \MT@warning@n1{%
2931 The #1 set '\MT@val' is undeclared.\MessageBreak
2932 Using set '\@nameuse{MT@\@tempb @setname}' instead}%
2933 }%
2934 }%
2935 }%
2936 }%
2937 }%
2938 }%
2939 }
2940 \MT@define@option@{spacing}
2941 \MT@define@option@{kerning}
2942 \MT@define@option@{tracking}

```

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

```

2943 \def\MT@def@bool@opt#1#2{%
2944 \define@key{MT}{#1}[]{%
2945 \MT@ifempty{#1}%
2946 {\def\@tempa{true}}%
2947 {\def\@tempa{#1}}%
2948 \MT@ifstreq\@tempa{true}\relax%
2949 \MT@ifstreq\@tempa{false}\relax%
2950 \MT@warning@n1{%
2951 ~#1' is not an admissible value for option\MessageBreak
2952 ~#1'. Assuming ~false'%
2953 \def\@tempa{false}%
2954 }%
2955 }%
2956 #2%
2957 }%
2958 }

```

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

```

2959 \def\MT@def@simple@bool@opt#1{\MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
2960 \MT@map@tlist@n{{draft}{auto}{selected}{babel}}\MT@def@simple@bool@opt

```

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

```

2961 \MT@def@bool@opt{DVIoutput}{%
2962 \csname if\@tempa\endcsname
2963 \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
2964 \pdfoutput\z@
2965 \else
2966 \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
2967 \pdfoutput\@ne
2968 \fi
2969 }

```

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.

```

2970 \MT@def@bool@opt{defersetup}{%
2971   \csname if\@tempa\endcsname \else
2972     \AtEndOfPackage{%
2973       \MT@setup@
2974       \let\MT@setup@\empty
2975       \let\MT@addto@setup\@firstofone
2976     }%
2977   \fi
2978 }

```

`final` is the opposite to `draft`.

```

2979 \MT@def@bool@opt{final}{%
2980   \csname if\@tempa\endcsname
2981     \MT@draftfalse
2982   \else
2983     \MT@drafttrue
2984   \fi
2985 }

```

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

```

2986 \define@key{MT}{verbose}[]{%
2987   \let\MT@vinfo\MT@info@n
2988   \MT@ifempty{#1}%
2989     {\def\@tempa{true}}%
2990     {\def\@tempa{#1}}%
2991   \MT@ifstreq\@tempa{true}\relax%

```

Take problems seriously.

```

2992   \MT@ifstreq\@tempa{errors}{%
2993     \let\MT@warning\MT@warn@err
2994     \let\MT@warning@n\MT@warn@err
2995   }{%
2996     \let\MT@vinfo\gobble
2997     \MT@ifstreq\@tempa{false}\relax%
2998     \MT@warning@n{%
2999       `#1' is not an admissible value for option\MessageBreak
3000       `verbose'. Assuming `false'}%
3001   }%
3002 }%
3003 }%
3004 }

```

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

```

3005 /package
3006 \def\MT@def@num@opt#1{%
3007   \define@key{MT}{#1}[]{%
3008     \MT@ifempty{##1}%
3009       {\MT@let@cn\@tempa{MT@#1@default}}%
3010       {\def\@tempa{##1}}%

```

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

```

3011   \MT@ifint\@tempa{%
3012     \MT@edef@n{MT@#1}{\@tempa}%
3013   }{\MT@warning@n{%
3014     Value `##1' for option `#1' is not a number.\MessageBreak
3015     Using default value of \number\@nameuse{MT@#1@default}}%
3016   }%

```

```

3017 }%
3018 }
3019 <package>\MT@map@tlist@n{{stretch}{shrink}{step}{letterspace}}\MT@def@num@opt
3020 <letterspace>\MT@def@num@opt{letterspace}
3021 <*package>

```

factor will define the protrusion factor only.

```

3022 \define@key{MT}{factor}[]{%
3023   \MT@ifempty{#1}%
3024     {\let\@tempa\MT@factor@default}%
3025     {\def\@tempa{#1}}}%
3026 \MT@ifint\@tempa{%
3027   \edef\MT@pr@factor{\@tempa}%
3028   }{\MT@warning@nl{%
3029     Value `#1' for option `factor' is not a number.\MessageBreak
3030     Using default value of \number\MT@factor@default}%
3031   }%
3032 }

```

Unit for protrusion codes.

```

3033 \define@key{MT}{unit}[]{%
3034   \MT@ifempty{#1}%
3035     {\def\@tempa{character}}%
3036     {\KV@sp@def\@tempa{#1}}}%
3037 \MT@ifstreq\@tempa{character}\relax{%
3038   \MT@ifdimen\@tempa{%
3039     \let\MT@pr@unit\@tempa
3040   }{%
3041     \MT@warning@nl{\@tempa' is not a dimension. Ignoring it and\MessageBreak
3042       setting values relative to character widths}%
3043   }%
3044 }%
3045 }

```

14.4.2 Reading the Configuration File

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

```

3046 \MT@protrusiontrue
3047 \ifnum\pdfoutput<\@ne \else

```

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

```

3048   \MT@requires@pdftex4{
3049     \MT@expansiontrue
3050     \MT@autottrue
3051   }\relax
3052 \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.

```

3053 \define@key{MT}{config}[]{\relax}
3054 \def\MT@get@config#1config=#2,#3\@nil{%
3055   \MT@ifempty{#2}%

```

```

3056 {\def\MT@config@file{\MT@MT.cfg}}%
3057 {\KV@esp@def\MT@config@file{#2.cfg}}%
3058 }
3059 \expandafter\expandafter\expandafter\MT@get@config
3060 \csname opt@\@currname.\@currentx\endcsname,config=,\@nil

```

Load the file.

```

3061 \IfFileExists{\MT@config@file}{%
3062 \MT@info@nl{Loading configuration file \MT@config@file}%
3063 \MT@begin@catcodes
3064 \let\MT@begin@catcodes\relax
3065 \let\MT@end@catcodes\relax
3066 \let\MT@curr@file\MT@config@file
3067 \input{\MT@config@file}%
3068 \endgroup
3069 }{\MT@warning@nl{%
3070 Could not find configuration file '\MT@config@file'!\MessageBreak
3071 This will almost certainly cause undesired results.\MessageBreak
3072 Please fix your installation}%
3073 }

```

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

```

3074 \MT@map@clist@c\MT@features{%
3075 \MT@ifdefined@n@TF{MT@default@#1@set}\relax
3076 {\MT@gdef@n{MT@default@#1@set}{all}}%
3077 }

```

`\MT@check@active@set` We have to make sure that font sets are active. If the user didn’t activate any, we use those sets declared by `\DeclareMicrotypeSetDefault` (this is done at the end of the preamble).

```

3078 \def\MT@check@active@set#1{%
3079 \MT@ifdefined@n@TF{MT@#1@setname}{%
3080 \MT@info@nl{Using \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}}%
3081 }{%
3082 \global\MT@let@nn{MT@#1@setname}{MT@default@#1@set}%
3083 \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}}%
3084 }
3085 }

```

14.4.3 Hook for Other Packages

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

This hook was needed in versions prior to 1.9a to overcome the situation that (1) the microtype package should be loaded after all font defaults have been set up (hence, using `\@ifpackageloaded` in the font package 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.

```

3086 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
3087   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3088   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
3089 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook

```

14.4.4 Changing Options Later

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

```

3090 \def\microtypesetup{\setkeys{MT}}
3091 \MT@addto@setup{\def\microtypesetup{\setkeys{MTX}}}
3092 \def\MT@define@optionX#1#2{%
3093   \define@key{MTX}{#1}[true]{%
3094     \KV@sp@def\@tempb{#1}%
3095     \MT@map@clist@n{#1}{%
3096       \KV@sp@def\MT@val{###1}%
3097       \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
3098       \MT@ifempty\MT@val\relax{%
3099         \@tempcnta=\m@ne
3100         \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.

```

3101     \MT@checksetup\@tempb{%
3102       \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
3103       \MT@info{Enabling #1
3104         (level \number\csname MT@\@tempb @level\endcsname)}%
3105     }%
3106   }{%
3107     \MT@ifstreq\MT@val{false}{%
3108       \@tempcnta=\z@
3109       \MT@info{Disabling #1}%
3110     }{%
3111       \MT@ifstreq\MT@val{compatibility}{%
3112         \MT@checksetup\@tempb{%
3113           \@tempcnta=\@ne
3114           \MT@let@nc{MT@\@tempb @level}\@ne
3115           \MT@info{Setting #1 to level 1}%
3116         }%
3117       }{%
3118         \MT@ifstreq\MT@val{nocompatibility}{%
3119           \MT@checksetup\@tempb{%
3120             \@tempcnta=\tw@
3121             \MT@let@nc{MT@\@tempb @level}\tw@
3122             \MT@info{Setting #1 to level 2}%
3123           }%
3124         }{%

```

```

3125         \MT@warning{%
3126             Value `~\MT@val' for key `~#1' not recognised.\MessageBreak
3127             Use any of `true', `false', `compatibility' or\MessageBreak
3128             `nocompatibility'%
3129         }%
3130     }%
3131 }%
3132 }%
3133 \ifnum\@tempcnta>\m@ne
3134 #2\@tempcnta\relax
3135 \fi
3136 }%
3137 }%
3138 }%
3139 }

```

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

```

3140 \def\MT@checksetup#1{%
3141     \expandafter\csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
3142     \expandafter\@firstofone
3143     \else
3144         \MT@warning{%
3145             You cannot enable \nameuse{MT@abbr@#1} if it was disabled\MessageBreak
3146             in the package options,%
3147             \expandafter\@gobble
3148         \fi
3149     }
3150 \MT@define@optionX{protrusion}\pdfprotrudechars
3151 \MT@define@optionX{expansion}\pdfadjustspacing

```

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

```

3152 \MT@requires@pdftex6{
3153     \def\MT@define@optionX@#1#2{%
3154         \define@key{MTX}{#1}[true]{%
3155             \KV@sp@def\@tempb{#1}%
3156             \MT@map@clist@{##1}{%
3157                 \KV@sp@def\MT@val{###1}%
3158                 \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
3159                 \MT@ifempty\MT@val\relax{%
3160                     \@tempcnta=\m@ne
3161                     \MT@ifstreq\MT@val{true}{%
3162                         \MT@checksetup\@tempb{%
3163                             \@tempcnta=\@ne
3164                             \MT@info{Enabling #1}%
3165                         }%
3166                     }{%
3167                         \MT@ifstreq\MT@val{false}{%
3168                             \@tempcnta=\z@
3169                             \MT@info{Disabling #1}%
3170                         }{%
3171                             \MT@warning{%
3172                                 Value `~\MT@val' for key `~#1' not recognised.\MessageBreak
3173                                 Use either `true' or `false'%
3174                             }%
3175                         }%
3176                     }
3177                 \ifnum\@tempcnta>\m@ne
3178                     #2\relax
3179                 \fi
3180             }%
3181         }%

```

```

3182 }
3183 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@
3184 \MT@trackingfalse\else\MT@trackingtrue\fi}
3185 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
3186 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3187 \pdfappendkern \@tempcnta}
3188 }{

```

Disable for older pdf_T_EX versions.

```

3189 \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
3190 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
3191 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
3192 }
3193 \define@key{MTX}{activate}[true]{%
3194 \setkeys{MTX}{protrusion={#1}}%
3195 \setkeys{MTX}{expansion={#1}}%
3196 }

```

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

```

3197 \let\MT@saved@setupfont\MT@setupfont
3198 \define@key{MTX}{disable}[]{%
3199 \MT@info{Inactivate `~\MT@MT' package}%
3200 \let\MT@setupfont\gobble
3201 }
3202 \define@key{MTX}{enable}[]{%
3203 \MT@info{Reactivate `~\MT@MT' package}%
3204 \let\MT@setupfont\MT@saved@setupfont
3205 }

```

14.4.5 Processing the Options

`\MT@ProcessOptionsWithKV` Parse options.

```

3206 /package
3207 \def\MT@ProcessOptionsWithKV#1{%
3208 \let\@tempc\relax
3209 \let\MT@temp\@empty
3210 \MT@map@c@list@c@c@classoptionslist{%
3211 \def\CurrentOption{#1}%
3212 \MT@ifdefined@n@T{KV#1@CurrentOption}{%
3213 \edef\MT@temp{\MT@temp,\CurrentOption,}%
3214 \@expandtwoargs\@removeelement\CurrentOption
3215 \@unusedoptionlist\@unusedoptionlist
3216 }%
3217 }%
3218 \edef\MT@temp{%
3219 \noexpand\setkeys{#1}{%
3220 \MT@temp\@optionlist{\@currname.\@current}%
3221 }%
3222 }%
3223 \MT@temp
3224 \AtEndOfPackage{\let\@unprocessedoptions\relax}%
3225 \let\CurrentOption\@empty
3226 }
3227 \MT@ProcessOptionsWithKV{MT}
3228 *package

```

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


```

3229 \MT@addto@setup{%
3230 \ifMT@draft

```

We disable most of what we've just defined in the 3230 lines above if we are running in draft mode.

```

3231 \MT@warning@nl{'draft' option active.\MessageBreak
3232             Disabling all micro-typographic extensions.\MessageBreak
3233             This might lead to different line and page breaks}
3234 \MT@protrusionfalse
3235 \MT@expansionfalse
3236 \MT@trackingfalse
3237 \MT@spacingfalse
3238 \MT@kerningfalse
3239 \MT@babelfalse
3240 \let\MT@setupfont\relax
3241 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
3242 \renewcommand*\UseMicrotypeSet[2] [] {}
3243 \renewcommand*\SetProtrusion[3] [] {}
3244 \renewcommand*\SetExpansion[3] [] {}
3245 \renewcommand*\SetTracking[3] [] {}
3246 \renewcommand*\SetExtraSpacing[3] [] {}
3247 \renewcommand*\SetExtraKerning[3] [] {}
3248 \renewcommand*\DeclareCharacterInheritance[3] [] {}
3249 \renewcommand*\DeclareMicrotypeAlias[2] {}
3250 \renewcommand*\LoadMicrotypeFile[1] {}
3251 \renewcommand*\microtypesetup[1] {}
3252 \renewcommand*\microtypecontext[1] {}
3253 \else

```

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

```

3254 \ifnum\pdfoutput<\@ne
3255     \ifMT@opt@expansion \else
3256         \MT@expansionfalse
3257     \fi
3258 \fi

```

pdf_T_EX 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!

```

3259 \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3260             \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%

```

Fix the font sets.

```

3261 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set

```

Protrusion.

```

3262 \ifMT@protrusion
3263     \edef\MT@active@features{\MT@active@features,pr}
3264     \pdfprotrudechars\MT@pr@level
3265     \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
3266               \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3267               factor: \number\MT@pr@factor\fi
3268               \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}
3269     \MT@check@active@set{pr}

```

```

3270 \else
3271   \let\MT@protrusion\relax
3272   \MT@info@n1{No character protrusion}
3273 \fi

```

Expansion.

```

3274 \ifMT@expansion

```

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

```

3275   \ifnum\MT@stretch=\m@ne
3276     \let\MT@stretch\MT@stretch@default
3277 \fi

```

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

```

3278   \ifnum\MT@shrink=\m@ne
3279     \ifnum\MT@stretch>\z@
3280       \let\MT@shrink\MT@stretch
3281     \else
3282       \let\MT@shrink\MT@shrink@default
3283     \fi
3284 \fi

```

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

```

3285   \ifnum\MT@step=\m@ne
3286     \ifnum\MT@stretch>\MT@shrink
3287       \ifnum\MT@shrink=\z@
3288         \@tempcnta=\MT@stretch
3289       \else
3290         \@tempcnta=\MT@shrink
3291       \fi
3292     \else
3293       \ifnum\MT@stretch=\z@
3294         \@tempcnta=\MT@shrink
3295       \else
3296         \@tempcnta=\MT@stretch
3297       \fi
3298     \fi
3299     \divide\@tempcnta 5\relax
3300   \else
3301     \@tempcnta=\MT@step
3302     \ifnum\@tempcnta=\z@
3303       \MT@warning@n1{The expansion step cannot be set to zero.\MessageBreak
3304         Setting it to one}
3305     \fi
3306   \fi
3307   \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3308   \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).

```

3309   \let\MT@auto\empty
3310   \ifMT@auto
3311     \MT@requires@pdftex4{%

```

We turn off automatic expansion if output mode is DVI.

```

3312     \ifnum\pdfoutput<\@ne
3313       \ifMT@opt@auto
3314         \MT@warning@n1{%

```

```

3315         Automatic font expansion only works for PDF output.\MessageBreak
3316         However, you are creating a DVI file. I will switch\MessageBreak
3317         automatic font expansion off and hope that expanded\MessageBreak
3318         fonts are available}
3319     \fi
3320     \MT@autofalse
3321 \else
3322     \def\MT@auto{autoexpand}
3323 \fi

```

Also, if pdf_TE_X is too old.

```

3324     }{%
3325     \MT@warning@nl{%
3326         The pdftex version you are using is too old for\MessageBreak
3327         automatic font expansion. I will switch it off and\MessageBreak
3328         hope that expanded fonts are available.\MessageBreak
3329         Otherwise, install pdftex version 1.20 or newer}
3330     \MT@autofalse
3331     \def\MT@auto{1000 }
3332 }

```

No automatic expansion.

```

3333 \else
3334     \ifnum\MT@pdftex@no < 4
3335     \def\MT@auto{1000 }
3336 \fi
3337 \fi

```

Choose the appropriate macro for selected expansion.

```

3338 \ifMT@selected
3339     \let\MT@set@ex@codes\MT@set@ex@codes@s
3340 \else
3341     \let\MT@set@ex@codes\MT@set@ex@codes@n
3342 \fi

```

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

```

3343 \ifnum\MT@stretch=\z@
3344     \ifnum\MT@shrink=\z@
3345     \MT@warning@nl{%
3346         Both the stretch and shrink limit are set to zero.\MessageBreak
3347         Disabling font expansion}
3348     \MT@expansionfalse
3349 \fi
3350 \fi
3351 \fi
3352 \ifMT@expansion
3353     \edef\MT@active@features{\MT@active@features,ex}%
3354     \pdfadjustspacing\MT@ex@level
3355     \MT@info@nl{\ifMT@auto\else Non-\fi Automatic font expansion enabled
3356         (level \number\MT@ex@level),\MessageBreak
3357         stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3358         step: \number\MT@step, \ifMT@selected\else non-\fi selected}
3359     \MT@check@active@set{ex}

```

Inside \showhyphens, font expansion should be disabled.

```

3360 \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3361     \color@begingroup\everypar{}\parfillskip\z@skip
3362     \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3363     \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

```

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

```

3364 \gdef\showhyphens#1{\setbox0\vbox{%
3365     \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip

```

```

3366      \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3367      \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

3368 \else
3369   \let\MT@expansion\relax
3370   \MT@info@nl{No font expansion}
3371 \fi
3372 }
3373 \MT@requires@pdftex6{
3374 </package>
3375 \MT@addto@setup{%
3376 <*package>

```

Tracking, spacing and kerning.

```

3377 \ifMT@tracking
3378   \edef\MT@active@features{\MT@active@features,tr}
3379   \MT@info@nl{Tracking enabled}
3380   \MT@check@active@set{tr}
3381 \else
3382   \let\MT@tracking\relax
3383   \MT@info@nl{No tracking}
3384 \fi
3385 \ifMT@spacing
3386   \edef\MT@active@features{\MT@active@features,sp}
3387   \pdfadjustinterwordglue\@ne
3388   \MT@info@nl{Adjustment of interword spacing enabled}
3389   \MT@check@active@set{sp}
3390 \else
3391   \let\MT@spacing\relax
3392   \MT@info@nl{No adjustment of interword spacing}
3393 \fi
3394 \ifMT@kerning
3395   \edef\MT@active@features{\MT@active@features,kn}
3396   \pdfprependkern\@ne
3397   \pdfappendkern\@ne
3398   \MT@info@nl{Adjustment of character kerning enabled}
3399   \MT@check@active@set{kn}
3400 \else
3401   \let\MT@kerning\relax
3402   \MT@info@nl{No adjustment of character kerning}
3403 \fi
3404 </package>
3405 \ifnum\MT@letterspace=\m@ne
3406   \let\MT@letterspace\MT@letterspace@default
3407 \else
3408   \MT@ls@too@large\MT@letterspace
3409 \fi
3410 }

```

If pdf_T_EX is too old, we disable tracking, spacing and kerning.

```

3411 <*package>
3412 {{
3413   \MT@addto@setup{%
3414     \ifMT@tracking
3415       \MT@warning@nl{Tracking only works with pdftex version 1.40\MessageBreak
3416         or newer. Switching it off}%
3417     \else
3418       \MT@info@nl{No tracking (pdftex too old)}
3419     \fi
3420     \MT@trackingfalse
3421     \let\MT@tracking\relax
3422     \ifMT@spacing
3423       \MT@warning@nl{Adjustment of interword spacing only works with\MessageBreak

```

```

3424     pdfTeX version 1.40 or newer. Switching it off}%
3425 \else
3426   \MT@info@nl{No adjustment of interword spacing (pdfTeX too old)}
3427 \fi
3428 \MT@spacingfalse
3429 \let\MT@spacing\relax
3430 \ifMT@kerning
3431   \MT@warning@nl{Character kerning only works with\MessageBreak
3432     pdfTeX version 1.40 or newer. Switching it off}%
3433 \else
3434   \MT@info@nl{No adjustment of character kerning (pdfTeX too old)}
3435 \fi
3436 \MT@kerningfalse
3437 \let\MT@kerning\relax
3438 }
3439 }

```

Warning if `\nonfrenchspacing` is active, since space factors will be ignored with `\pdfadjustinterwordglue>0`. Why 1500? Because some packages redefine `\frenchspacing`. See the c.t.t thread ‘`\frenchspacing` with AMS packages and babel’, started by Philipp Lehman on August 16, 2005: [<ddtbaj\\$rob\\$1@online.de>](mailto:ddtbajrob1@online.de).

```

3440 \MT@requires@pdfTeX6{
3441   \AtBeginDocument{%
3442     \ifMT@spacing
3443       \ifMT@babel \else
3444         \ifnum\sfcode~\> 1500
3445           \MT@ifstreql\MT@sp@context{nonfrench}\relax{%
3446             \MT@warning@nl{%
3447               \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3448               interword spacing will disable it. You might want\MessageBreak
3449               to add ~\@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
3450               to your preamble}%
3451           }%
3452         \fi
3453       \fi
3454     \fi
3455   }
3456 }\relax

```

`\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```

3457 \MT@requires@pdfTeX5{
3458   \MT@addto@setup{%
3459     \ifMT@noligatures \else
3460       \let\MT@noligatures\relax
3461     \fi
3462   }
3463 }\relax

```

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

```

3464 \MT@addto@setup{%
3465   \ifx\MT@active@features\@empty \else
3466     \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3467   \fi
3468   \MT@documenttrue
3469 }

```

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

```

3470 \def\MT@set@babel@context#1{%
3471   \MT@ifdefined@n@TF{MT@babel@#1}{%

```

```

3472 \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
3473 \expandafter\MT@exp@one@n\expandafter\microtypecontext
3474 \csname MT@babel@#1\endcsname
3475 }{%
3476 \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3477 }%
3478 }

```

\MT@shorthandoff Active characters can only be switched off if babel isn't loaded after microtype.

```

3479 \ifpackageloaded{babel}{
3480 \def\MT@shorthandoff#1#2{%
3481 \MT@info@n1{Switching off #1 babel's active characters}%
3482 \shorthandoff{#2}}
3483 }{
3484 \def\MT@shorthandoff#1#2{%
3485 \MT@error{You must load the `babel' package before microtype}
3486 {Otherwise, I cannot switch off the active characters
3487 for #1 babel.}}
3488 }

```

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

```

3489 \MT@addto@setup{%
3490 \ifMT@babel
3491 \ifpackageloaded{babel}{%
3492 \MT@info@n1{Redefining babel's language switching commands}
3493 \let\MT@orig@select@language\select@language
3494 \def\select@language#1{%
3495 \MT@orig@select@language{#1}%
3496 \MT@set@babel@context{#1}%
3497 }
3498 \let\MT@orig@foreign@language\foreign@language
3499 \def\foreign@language#1{%
3500 \MT@orig@foreign@language{#1}%
3501 \MT@set@babel@context{#1}%
3502 }

```

Disable French babel's active characters.

```

3503 \ifMT@kerning
3504 \MT@if@false
3505 \MT@with@babel@and{french} \MT@if@true
3506 \MT@with@babel@and{frenchb} \MT@if@true
3507 \MT@with@babel@and{français} \MT@if@true
3508 \MT@with@babel@and{acadian} \MT@if@true
3509 \MT@with@babel@and{canadien} \MT@if@true
3510 \ifMT@if@MT@shorthandoff{French}{;:!?}\fi

```

Disable Turkish babel's active characters.

```

3511 \MT@with@babel@and{turkish}{\MT@shorthandoff{Turkish}{:!=}}
3512 \fi

```

In case babel was loaded before microtype:

```

3513 \MT@set@babel@context\languagename
3514 }{%
3515 \MT@warning@n1{You did not load the babel package.\MessageBreak
3516 The `babel' option won't have any effect}
3517 }
3518 \fi
3519 }

```

Now we close the \fi from \ifMT@draft.

```

3520 \MT@addto@setup\fi

```

Set up the current font, most likely the normal font. This has to come after all of the setup (including anything from the preamble) has been dealt with.

```

3521 \AtBeginDocument\selectfont
\MT@curr@file This is the current file (hopefully with the correct extension).
3522 \edef\MT@curr@file{\jobname.tex}

That was that.
3523 </package>
3524 </package|letterspace>

```

15 Configuration Files

Let's now write the font configuration files.

```

3525 <*config>
3526

```

15.1 Font Sets

We first declare some sets in the main configuration file.

```

3527 <*m-t>
3528 %%% -----
3529 %%% FONT SETS
3530
3531 \DeclareMicrotypeSet{all}
3532 { }
3533
3534 \DeclareMicrotypeSet{allmath}
3535 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
3536
3537 \DeclareMicrotypeSet{alltext}
3538 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
3539
3540 \DeclareMicrotypeSet{basicmath}
3541 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
3542   family = {rm*,sf*},
3543   series  = {md*},
3544   size    = {normalsize,footnotesize,small,large}
3545 }
3546
3547 \DeclareMicrotypeSet{basictext}
3548 { encoding = {OT1,T1,LY1,OT4,QX,T5},
3549   family = {rm*,sf*},
3550   series  = {md*},
3551   size    = {normalsize,footnotesize,small,large}
3552 }
3553
3554 \DeclareMicrotypeSet{smallcaps}
3555 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3556   shape    = {sc}
3557 }
3558
3559 \DeclareMicrotypeSet{footnotesize}
3560 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3561   size     = {-small}
3562 }
3563
3564 \DeclareMicrotypeSet{scriptsize}

```

```

3565 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3566       size    = {-footnotesize}
3567 }
3568
3569 \DeclareMicrotypeSet{normalfont}
3570 { font = */*/*/*/* }
3571

```

The default sets.

```

3572 %%% -----
3573 %%% DEFAULT SETS
3574
3575 \DeclareMicrotypeSetDefault[protrusion]{alltext}
3576 \DeclareMicrotypeSetDefault[expansion]{basictext}
3577 \DeclareMicrotypeSetDefault[spacing]{basictext}
3578 \DeclareMicrotypeSetDefault[kerning]{alltext}
3579 \DeclareMicrotypeSetDefault[tracking]{normalfont}
3580

```

15.2 Font Aliases

Fonts that are ‘the same’.

```

3581 %%% -----
3582 %%% FONT ALIASES
3583

```

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.

```

3584 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
3585 \DeclareMicrotypeAlias{aer}{cmr} % ae
3586 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
3587 \DeclareMicrotypeAlias{cmor}{cmr} % eco
3588 \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.

```

3589 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
3590 \DeclareMicrotypeAlias{qpl}{ppl} % qfonts/QuasiPalatino
3591 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
3592 \DeclareMicrotypeAlias{qtm}{ptm} % qfonts/QuasiTimes

```

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

The eulervm package virtually extends the Euler fonts.

```

3593 \DeclareMicrotypeAlias{zeur}{eur} % Euler VM
3594 \DeclareMicrotypeAlias{zeus}{eus}

```

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

```

3595 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
3596 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond

```

Euro symbol fonts, to save some files.

```

3597 \DeclareMicrotypeAlias{zpeus}{zpeu} % Adobe Euro sans -> serif
3598 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
3599 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
3600

```


15.3 Interaction with babel

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

```

3601 %%% -----
3602 %%% INTERACTION WITH THE `babel' PACKAGE
3603
3604 \DeclareMicrotypeBabelHook
3605   {french,français,acadian,canadien}
3606   {kerning=french, spacing=}
3607
3608 \DeclareMicrotypeBabelHook
3609   {english,UKenglish,british,american,USenglish}
3610   {kerning=, spacing=nonfrench}
3611
3612 \DeclareMicrotypeBabelHook
3613   {turkish}
3614   {kerning=turkish, spacing=}
3615

```

15.4 Note on Admissible Characters

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

```

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

```

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

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

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

15.5 Character Inheritance

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

```

3616 </m-t>
3617 <{*m-t|zpeu|mvs}>
3618 %%% -----
3619 %%% CHARACTER INHERITANCE
3620
3621 </m-t|zpeu|mvs>
3622 <{*m-t}>

```

15.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 (‘fi’ ligature), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```
3623 \DeclareCharacterInheritance
3624 { encoding = OT1 }
3625 { f = {011}, % ff
3626     i = {\i},
3627     j = {\j},
3628     O = {\O},
3629     o = {\o},
3630 }
3631
```

15.5.2 T1

Candidates here: 028 (‘fi’), 029 (‘fl’), 030 (‘ffi’), 031 (‘ffl’), 156 (‘IJ’ ligature, since L^AT_EX 2005/12/01 accessible as \IJ), 188 (‘ij’, \i j), Æ, æ, Œ, œ.

```
3632 \DeclareCharacterInheritance
3633 { encoding = T1 }
3634 { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
3635     a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
3636     C = {\`C,\'C,\^C,\~C,\"C,\r C,\k C,\u C},
3637     c = {\`c,\'c,\^c,\~c,\"c,\r c,\k c,\u c},
3638     D = {\`D,\'D,\^D,\~D,\"D,\r D,\k D,\u D},
3639     d = {\`d,\'d,\^d,\~d,\"d,\r d,\k d,\u d},
3640     E = {\`E,\'E,\^E,\~E,\"E,\r E,\k E,\u E},
3641     e = {\`e,\'e,\^e,\~e,\"e,\r e,\k e,\u e},
3642     f = {027}, % ff
3643     G = {\u G},
3644     g = {\u g},
3645     I = {\`I,\'I,\^I,\~I,\"I,\r I,\k I,\u I},
3646     i = {\`i,\'i,\^i,\~i,\"i,\r i,\k i,\u i},
3647     j = {\j},
3648     L = {\`L,\'L,\^L,\~L,\"L,\r L,\k L,\u L},
3649     l = {\`l,\'l,\^l,\~l,\"l,\r l,\k l,\u l},
3650     N = {\`N,\'N,\^N,\~N,\"N,\r N,\k N,\u N},
3651     n = {\`n,\'n,\^n,\~n,\"n,\r n,\k n,\u n},
3652     O = {\`O,\'O,\^O,\~O,\"O,\r O,\k O,\u O},
3653     o = {\`o,\'o,\^o,\~o,\"o,\r o,\k o,\u o},
3654     R = {\`R,\'R,\^R,\~R,\"R,\r R,\k R,\u R},
3655     r = {\`r,\'r,\^r,\~r,\"r,\r r,\k r,\u r},
3656     S = {\`S,\'S,\^S,\~S,\"S,\r S,\k S,\u S},
3657     s = {\`s,\'s,\^s,\~s,\"s,\r s,\k s,\u s},
3658     T = {\`T,\'T,\^T,\~T,\"T,\r T,\k T,\u T},
3659     t = {\`t,\'t,\^t,\~t,\"t,\r t,\k t,\u t},
3660     U = {\`U,\'U,\^U,\~U,\"U,\r U,\k U,\u U},
3661     u = {\`u,\'u,\^u,\~u,\"u,\r u,\k u,\u u},
3662     Y = {\`Y,\'Y,\^Y,\~Y,\"Y,\r Y,\k Y,\u Y},
3663     y = {\`y,\'y,\^y,\~y,\"y,\r y,\k y,\u y},
3664     Z = {\`Z,\'Z,\^Z,\~Z,\"Z,\r Z,\k Z,\u Z},
3665     z = {\`z,\'z,\^z,\~z,\"z,\r z,\k z,\u z},
3666     - = {127},
3667 }
3668
```

15.5.3 LY1

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

```

3669 \DeclareCharacterInheritance
3670 { encoding = LY1 }
3671 { A = {\`A,\`A,\^A,\~A,\"A,\"r A},
3672   a = {\`a,\`a,\^a,\~a,\"a,\"r a},
3673   C = {\c C},
3674   c = {\c c},
3675   D = {\DH},
3676   E = {\`E,\`E,\^E,\"E},
3677   e = {\`e,\`e,\^e,\"e},
3678   f = {011}, % ff
3679   I = {\`I,\`I,\^I,\"I},
3680   i = {\`i,\`i,\^i,\"i,\"i},
3681   L = {\L},
3682   l = {\l},
3683   N = {\~N},
3684   n = {\~n},
3685   O = {\`O,\`O,\^O,\~O,\"O,\"O},
3686   o = {\`o,\`o,\^o,\~o,\"o,\"o},
3687   S = {\v S},
3688   s = {\v s},
3689   U = {\`U,\`U,\^U,\"U},
3690   u = {\`u,\`u,\^u,\"u},
3691   Y = {\`Y,\"Y},
3692   y = {\`y,\"y},
3693   Z = {\v Z},
3694   z = {\v z},
3695 }
3696

```

15.5.4 OT4

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

```

3697 \DeclareCharacterInheritance
3698 { encoding = OT4 }
3699 { A = {\k A},
3700   a = {\k a},
3701   C = {\`C},
3702   c = {\`c},
3703   E = {\k E},
3704   e = {\k e},
3705   f = {011}, % ff
3706   i = {\i},
3707   j = {\j},
3708   L = {\L},
3709   l = {\l},
3710   N = {\`N},
3711   n = {\`n},
3712   O = {\O,\"O},
3713   o = {\o,\"o},
3714   S = {\`S},
3715   s = {\`s},
3716   Z = {\`Z,\"Z},
3717   z = {\`z,\"z},
3718 }
3719

```

15.5.5 QX

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

```

3720 \DeclareCharacterInheritance
3721 { encoding = QX }
3722 { A = {\`A,\'A,\^A,\~A,\"A,\k A,\AA},
3723   a = {\`a,\'a,\^a,\~a,\"a,\k a,\aa},
3724   C = {\`C,\c C},
3725   c = {\`c,\c c},
3726   D = {\DH},
3727   E = {\`E,\'E,\^E,\"E,\k E},
3728   e = {\`e,\'e,\^e,\"e,\k e},
3729   f = {011}, % ff
3730   I = {\`I,\'I,\^I,\"I,\k I},
3731   i = {\`i,\'i,\^i,\"i,\k i,\i},
3732   j = {\j},
3733   L = {\L},
3734   l = {\l},
3735   N = {\`N,\~N},
3736   n = {\`n,\~n},
3737   O = {\0,\`0,\'0,\^0,\~0,\"0},
3738   o = {\`o,\'o,\^o,\~o,\"o},
3739   S = {\`S,\c S,\v S},
3740   s = {\`s,\c s,\v s},
3741   T = {\c T},
3742   t = {\c t},
3743   U = {\`U,\'U,\^U,\"U,\k U},
3744   u = {\`u,\'u,\^u,\"u,\k u},
3745   Y = {\`Y,\"Y},
3746   y = {\`y,\"y},
3747   Z = {\`Z,\"Z,\v Z},
3748   z = {\`z,\"z,\v z},
3749   . = \textellipsis
3750 }
3751
```

15.5.6 T5

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

```

3752 \DeclareCharacterInheritance
3753 { encoding = T5 }
3754 { A = {\`A,\'A,\~A,\h A,\d A,\^A,\u A,
3755   \\\Acircumflex,\'\Acircumflex,\-\Acircumflex,\h\Acircumflex,\d\Acircumflex,
3756   \\\Abreve,\'\Abreve,\-\Abreve,\h\Abreve,\d\Abreve},
3757   a = {\`a,\'a,\~a,\h a,\d a,\^a,\u a,
3758   \\\acircumflex,\'\acircumflex,\-\acircumflex,\h\acircumflex,\d\acircumflex,
3759   \\\abreve,\'\abreve,\-\abreve,\h\abreve,\d\abreve},
3760   D = {\DJ},
3761   d = {\dj},
3762   E = {\`E,\'E,\~E,\h E,\d E,\^E,
3763   \\\Ecircumflex,\'\Ecircumflex,\-\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
3764   e = {\`e,\'e,\~e,\h e,\d e,\^e,
3765   \\\ecircumflex,\'\ecircumflex,\-\ecircumflex,\h\ecircumflex,\d\ecircumflex},
3766   I = {\`I,\'I,\~I,\h I,\d I},
3767   i = {\`i,\'i,\~i,\h i,\d i,\i},
3768   O = {\`O,\'O,\~O,\h O,\d O,\^O,\horn O,
3769   \\\Ocircumflex,\'\Ocircumflex,\-\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
3770   \\\Ohorn,\'\Ohorn,\~Ohorn,\h\Ohorn,\d\Ohorn},

```

```

3771 o = {\`o,\`o,\~o,\h o,\d o,\^o,\horn o,
3772         \`\ocircumflex,\`\ocircumflex,\~\ocircumflex,\h\ocircumflex,\d\ocircumflex,
3773         \`\ohorn,\`\ohorn,\~\ohorn,\h\ohorn,\d\ohorn},
3774 U = {\`U,\`U,\~U,\h U,\d U,\horn U,
3775         \`\Uhorn,\`\Uhorn,\~\Uhorn,\h\Uhorn,\d\Uhorn},
3776 u = {\`u,\`u,\~u,\h u,\d u,\horn u,
3777         \`\uhorn,\`\uhorn,\~\uhorn,\h\uhorn,\d\uhorn},
3778 Y = {\`Y,\`Y,\~Y,\h Y,\d Y},
3779 y = {\`y,\`y,\~y,\h y,\d y},
3780 }
3781
3782 </m-t>

```

15.5.7 Euro symbols

Make Euro symbols settings simpler.

```

3783 < *zpeu >
3784 \DeclareCharacterInheritance
3785 { encoding = U,
3786   family   = {zpeu,zpeus,eurosans} }
3787 { E = 128 }
3788
3789 </zpeu >
3790 < *mvs >
3791 \DeclareCharacterInheritance
3792 { encoding = OT1,
3793   family   = mvs }
3794 { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
3795

```

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.

```

3796 \DeclareCharacterInheritance
3797 { encoding = U,
3798   family   = mvs }
3799 { 164 = {099,100,101} }
3800
3801 </mvs >

```

15.6 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).

```

3802 < *m-t >
3803 %%% -----
3804 %%% EXPANSION SETTINGS
3805
3806 \SetExpansion
3807 [ name      = default      ]
3808 { encoding = {OT1,OT4,QX,T1,LY1} }
3809 {
3810   A = 500,      a = 700,
3811   \AE = 500,    \ae = 700,
3812   B = 700,      b = 700,
3813   C = 700,      c = 700,
3814   D = 500,      d = 700,
3815   E = 700,      e = 700,
3816   F = 700,

```

```

3817     G = 500,      g = 700,
3818     H = 700,      h = 700,
3819     K = 700,      k = 700,
3820     M = 700,      m = 700,
3821     N = 700,      n = 700,
3822     O = 500,      o = 700,
3823     \OE = 500,     \oe = 700,
3824     P = 700,      p = 700,
3825     Q = 500,      q = 700,
3826     R = 700,
3827     S = 700,      s = 700,
3828     U = 700,      u = 700,
3829     W = 700,      w = 700,
3830     Z = 700,      z = 700,
3831     2 = 700,
3832     3 = 700,
3833     6 = 700,
3834     8 = 700,
3835     9 = 700,
3836   }
3837

```

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

```

3838 \SetExpansion
3839   [ name      = T5 ]
3840   { encoding = T5 }
3841   {
3842     A = 500,      a = 700,
3843     B = 700,      b = 700,
3844     C = 700,      c = 700,
3845     D = 500,      d = 700,
3846     E = 700,      e = 700,
3847     F = 700,
3848     G = 500,      g = 700,
3849     H = 700,      h = 700,
3850     K = 700,      k = 700,
3851     M = 700,      m = 700,
3852     N = 700,      n = 700,
3853     O = 500,      o = 700,
3854     P = 700,      p = 700,
3855     Q = 500,      q = 700,
3856     R = 700,
3857     S = 700,      s = 700,
3858     U = 700,      u = 700,
3859     W = 700,      w = 700,
3860     Z = 700,      z = 700,
3861     2 = 700,
3862     3 = 700,
3863     6 = 700,
3864     8 = 700,
3865     9 = 700,
3866   }
3867
3868 </m-t>

```

15.7 Character Protrusion

```

3869 %%% -----
3870 %%% PROTRUSION SETTINGS
3871

```

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

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

15.7.1 Default

The default settings always use the most moderate value.

```
3872 <*cfg-t>
3873 \SetProtrusion
3874 <m-t> [ name      = default ]
```

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

```
3875 <bch> [ name      = bch-default ]
```

Computer Modern Roman (`cmr`),

```
3876 <cmr> [ name      = cmr-default ]
```

Adobe Garamond (`pad`, `padx`, `padj`),

```
3877 <pad> [ name      = pad-default ]
```

Minion¹³ (`pmnx`, `pmnj`),

```
3878 <pmn> [ name      = pmnj-default ]
```

Palatino (`ppl`, `pplx`, `pplj`),

```
3879 <ppl> [ name      = ppl-default ]
```

Times (`ptm`, `ptmx`, `ptmj`),

```
3880 <ptm> [ name      = ptm-default ]
```

and URW Garamond (`ugm`).

```
3881 <ugm> [ name      = ugm-default ]
```

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

```

3882 <m-t> { encoding = OT1      }
3883 <cmr> { }
3884 <bch|pad|pmn|ugm> { encoding = OT1,
3885 <ppl|ptm> { encoding = {OT1,OT4},
3886 <bch> family = bch }
3887 <pad> family = {pad,padx,padj} }
3888 <pmn> family = pmnj }
3889 <ppl> family = {ppl,pplx,pplj} }
3890 <ptm> family = {ptm,ptmx,ptmj} }
3891 <ugm> family = ugm }
3892 {
3893 <m-t|bch|cmr|pad|pmn|ppl|ptm> A = {50,50},
3894 <ugm> A = {50,100},
3895 <m-t|pad|ptm> \AE = {50, },
3896 <ugm> \AE = {150,50},
3897 <ugm> B = { ,50},
3898 <bch|pad|pmn|ugm> C = {50, },
3899 <bch|pad|pmn> D = { ,50},
3900 <ugm> D = { ,70},
3901 <ugm> E = { ,50},
3902 <m-t|bch|cmr|pad|pmn|ptm> F = { ,50},
3903 <ugm> F = { ,70},
3904 <bch|pad|pmn> G = {50, },
3905 <ugm> G = {50,50},
3906 <m-t|cmr|pad|pmn|ppl|ptm|ugm> J = {50, },
3907 <bch> J = {100, },
3908 K = { ,50},
3909 <m-t|bch|cmr|pad|pmn|ppl> L = { ,50},
3910 <ptm> L = { ,80},
3911 <ugm> L = { ,120},
3912 <bch|pad|pmn|ugm> O = {50,50},
3913 <pad|pmn> \OE = {50, },
3914 <ugm> \OE = {50,50},
3915 <ugm> P = { ,50},
3916 <bch|pad|pmn> Q = {50,70},
3917 <ugm> Q = {50,50},
3918 <bch> R = { ,50},
3919 <ugm> R = { ,70},
3920 <m-t|bch|cmr|pad|pmn|ppl|ptm> T = {50,50},
3921 <ugm> T = {70,70},
3922 <m-t|bch|cmr|pad|pmn|ppl|ptm> V = {50,50},
3923 <ugm> V = {70,70},
3924 <m-t|bch|cmr|pad|pmn|ppl|ptm> W = {50,50},
3925 <ugm> W = {70,70},
3926 <m-t|bch|cmr|pad|pmn|ppl|ptm> X = {50,50},
3927 <ugm> X = {50,70},
3928 <m-t|bch|cmr|pad|pmn|ppl> Y = {50,50},
3929 <ptm|ugm> Y = {80,80},
3930 <ugm> Z = {50,50},
3931 <m-t|bch|cmr|pad|pmn|ppl|ptm> k = { ,50},
3932 <ugm> k = { ,70},
3933 <pmn> l = { , -50},
3934 <pad|ppl> p = {50,50},
3935 <ugm> p = { ,50},
3936 <pad|ppl> q = {50, },
3937 r = { ,50},
3938 <cmr|pad|pmn> t = { ,70},
3939 <bch> t = { ,50},
3940 <ugm> t = { ,100},
3941 <m-t|bch|cmr|pad|pmn|ppl|ptm> v = {50,50},
3942 <ugm> v = {50,70},
3943 <m-t|bch|cmr|pad|pmn|ppl|ptm> w = {50,50},
3944 <ugm> w = {50,70},

```



```

3945      x = {50,50},
3946 <m-t|bch|pad|pmn>      y = { ,50},
3947 <cmr|ppl|ptm>      y = {50,70},
3948 <ugm>      y = { ,70},

3949 <cmr>      0 = { ,50},
3950 <m-t>      1 = {50,50},
3951 <bch|pad|ptm|ugm>      1 = {150,150},
3952 <cmr>      1 = {100,200},
3953 <pmn>      1 = { ,50},
3954 <ppl>      1 = {100,100},
3955 <bch|cmr|pad|ugm>      2 = {50,50},
3956 <cmr|pad|ugm>      3 = {50,50},
3957 <bch|pmn>      3 = {50, },
3958 <m-t|pad>      4 = {50,50},
3959 <bch>      4 = {100,50},
3960 <cmr|ugm>      4 = {70,70},
3961 <pmn>      4 = {50, },
3962 <ptm>      4 = {70, },
3963 <cmr>      5 = { ,50},
3964 <pad>      5 = {50,50},
3965 <bch>      6 = {50, },
3966 <cmr>      6 = { ,50},
3967 <pad>      6 = {50,50},
3968 <m-t>      7 = {50,50},
3969 <bch|pad|pmn|ugm>      7 = {50,80},
3970 <cmr|ptm>      7 = {50,100},
3971 <ppl>      7 = { ,50},
3972 <cmr>      8 = { ,50},
3973 <bch|pad>      9 = {50,50},
3974 <cmr>      9 = { ,50},
3975 <m-t|cmr|pad|pmn|ppl|ptm|ugm>      . = { ,700},
3976 <bch>      . = { ,600},
3977 {,}= { ,500},
3978 <m-t|cmr|pad|pmn|ppl|ptm|ugm>      : = { ,500},
3979 <bch>      : = { ,400},
3980 <m-t|bch|pad|pmn|ptm>      ; = { ,300},
3981 <cmr|ppl>      ; = { ,500},
3982 <ugm>      ; = { ,400},
3983      ! = { ,100},
3984 <m-t|pad|pmn|ptm>      ? = { ,100},
3985 <bch|cmr|ppl|ugm>      ? = { ,200},
3986 <pmn>      " = {300,300},
3987 <m-t|bch|cmr|pad|pmn|ppl>      @ = {50,50},
3988 <ptm>      @ = {100,100},
3989 <m-t|bch|cmr|pad|pmn|ppl|ptm>      ~ = {200,250},
3990 <ugm>      ~ = {300,350},
3991 <pad|ppl|ptm>      & = {50,100},
3992 <ugm>      & = { ,100},
3993 <m-t|cmr|pad|pmn>      \% = {50,50},
3994 <bch>      \% = { ,50},
3995 <ppl|ptm>      \% = {100,100},
3996 <ugm>      \% = {50,100},
3997 <m-t|ppl|ptm|ugm>      * = {200,200},
3998 <bch|pmn>      * = {200,300},
3999 <cmr|pad>      * = {300,300},
4000 <m-t|cmr|ppl|ptm>      + = {250,250},
4001 <bch>      + = {150,250},
4002 <pad>      + = {300,300},
4003 <pmn>      + = {150,200},
4004 <ugm>      + = {250,300},
4005 <ugm>      {=} = {200,200},
4006 <m-t|pad|pmn|ptm>      ( = {100, }, ) = { ,200},

```

```

4007 <bch|ugm>      ( = {200,  },    ) = {    ,200},
4008 <cmr|ppl>      ( = {100,  },    ) = {    ,300},
4009 <bch|pmn>      [ = {100,  },    ] = {    ,100},

4010 <m-t|pad|pmn|ptm> / = {100,200},
4011 <bch>          / = {    ,200},
4012 <cmr|ppl>      / = {200,300},
4013 <ugm>          / = {100,300},
4014 <m-t|ptm>      - = {500,500},
4015 <bch|cmr|ppl>  - = {400,500},
4016 <pad>          - = {300,500},
4017 <pmn>          - = {200,400},
4018 <ugm>          - = {500,600},
4019 <m-t|pmn>      \textendash      = {200,200},    \textendash      = {150,150},
4020 <bch>          \textendash      = {200,300},    \textendash      = {150,250},
4021 <cmr>          \textendash      = {400,300},    \textendash      = {300,200},
4022 <pad|ppl|ptm> \textendash      = {300,300},    \textendash      = {200,200},
4023 <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).

```

4024 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
4025 <cmr>         \textquoteleft = {500,700}, \textquoteright = {500,600},
4026 <pad|ppl>     \textquoteleft = {500,700}, \textquoteright = {500,700},
4027 <ptm>         \textquoteleft = {500,500}, \textquoteright = {300,500},
4028 <ugm>         \textquoteleft = {300,600}, \textquoteright = {300,600},
4029 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300},
4030 <cmr>         \textquotedblleft = {500,300}, \textquotedblright = {200,600},
4031 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400},
4032 <ugm>         \textquotedblleft = {400,400}, \textquotedblright = {400,400},
4033     }
4034

```

Greek uppercase letters are in OT1 encoding only.

```

4035 <*cmr>
4036 \SetProtrusion
4037 [ name      = cmr-OT1,
4038   load      = cmr-default ]
4039 { encoding = {OT1,OT4},
4040   family   = cmr   }
4041 {
4042   \AE = { 50,  },
4043   "00 = {    ,150}, % \Gamma
4044   "01 = {100,100}, % \Delta
4045   "02 = { 50, 50}, % \Theta
4046   "03 = {100,100}, % \Lambda
4047   "06 = { 50, 50}, % \Sigma
4048   "07 = {100,100}, % \Upsilon
4049   "08 = { 50, 50}, % \Phi
4050   "09 = { 50, 50}, % \Psi

```

Remaining slots can be found in the source file.

```

4051   }
4052
4053 </cmr>

```

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

```

4054 \SetProtrusion
4055 <m-t> [ name      = T1-default,
4056 <bch> [ name      = bch-T1,
4057 <cmr> [ name      = cmr-T1,

```

```

4058 <pad> [ name = pad-T1,
4059 <pmn> [ name = pmnj-T1,
4060 <ppl> [ name = ppl-T1,
4061 <ptm> [ name = ptm-T1,
4062 <ugm> [ name = ugm-T1,
4063 <m-t> load = default ]
4064 <bch> load = bch-default ]
4065 <cmr> load = cmr-default ]
4066 <pad> load = pad-default ]
4067 <pmn> load = pmnj-default ]
4068 <ppl> load = ppl-default ]
4069 <ptm> load = ptm-default ]
4070 <ugm> load = ugm-default ]
4071 <m-t> { encoding = {T1,LY1} }
4072 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4073 <ptm|ugm> { encoding = {T1},
4074 <bch> family = bch }
4075 <cmr> family = cmr }
4076 <pad> family = {pad,padx,padj} }
4077 <pmn> family = pmnj }
4078 <ppl> family = {ppl,pplx,pplj} }
4079 <ptm> family = {ptm,ptmx,ptmj} }
4080 <ugm> family = ugm }
4081 {
4082 <cmr> \AE = {50, },
4083 <bch> \OE = {50, },
4084 <pmn> \TH = { ,50},
4085 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
4086 <cmr> _ = {200,200},
4087 <ugm> _ = {100,200},
4088 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
4089 <bch> \textbackslash = {150,200},
4090 <cmr|ppl> \textbackslash = {200,300},
4091 <ugm> \textbackslash = {100,300},
4092 <ugm> \textbar = {200,200},
4093 <cmr> \textquotedblleft = {200,600},
4094 <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.

```

4095 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4096 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4097 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4098 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4099 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4100 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
4101 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
4102 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
4103 <pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
4104 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4105 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
4106 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4107 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
4108 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
4109 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200},
4110 <pmn> \textless = {100, }, \textgreater = { ,100},
4111 <pmn> \textvisiblespace = {100,100}, % not in LY1
4112 }
4113

```

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

```

4114 <*cmr>
4115 \SetProtrusion
4116 [ name      = lmr-T1,
4117   load      = cmr-T1 ]
4118 { encoding = {T1,LY1},
4119   family   = lmr      }
4120 {
4121   \textquotedblleft = {500,300},
4122   \quotedblbase     = {500,300},
4123 }
4124
4125 </cmr>

```

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

```

4126 <*-t|ptm>
4127 \SetProtrusion
4128 <m-t> [ name      = QX-default,
4129 <ptm>  [ name      = ptm-QX,
4130 <m-t>   load      = default ]
4131 <ptm>   load      = ptm-default ]
4132 <m-t> { encoding = QX }
4133 <ptm> { encoding = QX,
4134 <ptm>   family   = {ptm,ptmx,ptmj} }
4135 {
4136 <ptm>   * = {200,200},
4137   {=} = {100,100},
4138   \textunderscore = {100,100},
4139   \textbackslash   = {100,200},
4140   \quotedblbase    = {400,400},
4141 <m-t>   \guillemotleft = {200,200}, \guillemotright = {200,200},
4142 <ptm>   \guillemotleft = {300,300}, \guillemotright = {200,400},
4143   \textexclamdown = {100, }, \textquestiondown = {100, },
4144 <m-t>   \textbraceleft = {400,200}, \textbraceright = {200,400},
4145 <ptm>   \textbraceleft = {200,200}, \textbraceright = {200,300},
4146   \textless        = {200,100}, \textgreater       = {100,200},
4147   \textminus       = {200,200}, \textdegree      = {300,300},
4148 <m-t>   \copyright    = {100,100}, \textregistered = {100,100},
4149 <ptm>   \copyright    = {100,150}, \textregistered = {100,150},
4150 <ptm>   \textxgeq     = { ,100}, \textxleq      = {100, },
4151 <ptm>   \textalpha    = { , 50}, \textDelta    = { 70, 70},
4152 <ptm>   \textpi       = { 50, 80}, \textSigma    = { , 70},
4153 <ptm>   \textmu       = { , 80}, \texteuro      = { 50, 50},
4154 <ptm>   \textellipsis = {150,200}, \textasciitilde = { 80, 80},
4155 <ptm>   \textapprox    = { 50, 50}, \textinfty    = {100,100},
4156 <ptm>   \textdagger    = {150,150}, \textdaggerdbl = {100,100},
4157 <ptm>   \textdiv       = { 50,150}, \textsection  = { 80, 80},
4158 <ptm>   \texttimes     = {100,150}, \textpm       = { 50, 80},
4159 <ptm>   \textbullet    = {150,150}, \textperiodcentered = {300,300},
4160 <ptm>   \textquotesingle = {500,500}, \textquotedbl = {300,300},
4161 <ptm>   \textperthousand = { ,50},
4162 }
4163
4164 </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.

```

4165 <*cmr|bch>
4166 \SetProtrusion

```

```

4167 <cmr> [ name      = cmr-T5,
4168 <cmr>    load      = cmr-default ]
4169 <bch> [ name      = bch-T5,
4170 <bch>    load      = bch-default ]
4171 { encoding = T5,
4172 <cmr>    family   = cmr }
4173 <bch>    family   = bch }
4174 {
4175 <bch>    _ = {100,100},
4176 <bch>    \textbackslash = {150,200},
4177 <cmr>    \textbackslash = {200,300},
4178 <cmr>    \textquotedblleft = {200,600},
4179 <cmr>    \textquotedbl = {300,300},
4180 <bch>    \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4181 <cmr>    \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4182 <bch>    \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4183 <cmr>    \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4184 <bch>    \guillemotleft = {200,200}, \guillemotright = {150,300},
4185 <cmr>    \guillemotleft = {300,200}, \guillemotright = {100,400},
4186 <bch>    \textbraceleft = {200, }, \textbraceright = { ,300},
4187 <cmr>    \textbraceleft = {400,200}, \textbraceright = {200,400},
4188 \textless = {200,100}, \textgreater = {100,200},
4189 }
4190
4191 </cmr|bch>
4192 <*pmn>
4193 \SetProtrusion
4194 [ name      = pmnx-OT1,
4195   load      = pmnj-default ]
4196 { encoding = OT1,
4197   family   = pmnx }
4198 {
4199   1 = {230,180},
4200 }
4201
4202 \SetProtrusion
4203 [ name      = pmnx-T1,
4204   load      = pmnj-T1 ]
4205 { encoding = {T1,LY1},
4206   family   = pmnx }
4207 {
4208   1 = {230,180},
4209 }
4210
4211 </pmn>

```

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

```

4212 <*ptm>
4213 \SetProtrusion
4214 [ name      = ptm-LY1,
4215   load      = ptm-T1 ]
4216 { encoding = LY1,
4217   family   = {ptm,ptmx,ptmj} }
4218 {
4219   _ = {100,100},
4220   \texttrademark = {100,100},
4221   \textregistered = {100,100},
4222   \textcopyright = {100,100},
4223   \textdegree = {300,300},
4224   \textminus = {200,200},
4225   \textellipsis = {150,200},

```

```

4226 \texteuro           = { , }, % ?
4227 \textcent           = {100,100},
4228 \textquotesingle   = {500,500},
4229 \textflorin        = { 50, 70},
4230 \textdagger         = {150,150},
4231 \textdaggerdbl     = {100,100},
4232 \textperthousand   = { , 50},
4233 \textbullet         = {150,150},
4234 \textonesuperior    = {100,100},
4235 \texttwosuperior    = { 50, 50},
4236 \textthreesuperior = { 50, 50},
4237 \textperiodcentered = {300,300},
4238 \textplusminus      = { 50, 80},
4239 \textmultiply       = {100,100},
4240 \textdivide        = { 50,150},

```

Remaining slots in the source file.

```

4241 }
4242
4243 </ptm>

```

15.7.2 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.

```

4244 \SetProtrusion
4245 <m-t> [ name = OT1-it ]
4246 <bch> [ name = bch-it ]
4247 <cmr> [ name = cmr-it ]
4248 <pad> [ name = pad-it ]
4249 <pmn> [ name = pmnj-it ]
4250 <ppl> [ name = ppl-it ]
4251 <ptm> [ name = ptm-it ]
4252 <ugm> [ name = ugm-it ]
4253 <m-t|bch|pad|pmn|ugm> { encoding = OT1,
4254 <ppl|ptm> { encoding = {OT1,OT4},
4255 <bch> family = bch,
4256 <pad> family = {pad,padx,padj},
4257 <pmn> family = pmnj,
4258 <ppl> family = {ppl,pplx,pplj},
4259 <ptm> family = {ptm,ptmx,ptmj},
4260 <ugm> family = ugm,
4261 <! (cmr|ugm)> shape = {it,sl} }
4262 <ugm> shape = it }
4263 <cmr> { }
4264 {
4265 <cmr|ptm> A = {100,50},
4266 <pad|pmn> A = {50, },
4267 <ugm> A = { ,150},
4268 <ppl> A = {50,50},
4269 <ptm> \AE = {100, },
4270 <pad|ppl> \AE = {50, },
4271 <pmn> \AE = { , -50},
4272 <cmr|pad|ppl|ptm> B = {50, },
4273 <pmn> B = {20,-50},
4274 <bch|ppl|ptm|ugm> C = {50, },
4275 <cmr|pad> C = {100, },
4276 <pmn> C = {50,-50},
4277 <cmr|pad|ppl|ptm> D = {50,50},
4278 <pmn> D = {20, },

```

```

4279 <cmr|pad|ppl|ptm>    E = {50, },
4280 <pmn>                E = {20,-50},
4281 <cmr|pad|ptm>        F = {100, },
4282 <pmn>                F = {10, },
4283 <ppl>                 F = {50, },
4284 <bch|ppl|ptm|ugm>    G = {50, },
4285 <cmr|pad>            G = {100, },
4286 <pmn>                G = {50,-50},
4287 <cmr|pad|ppl|ptm>    H = {50, },
4288 <cmr|pad|ptm>        I = {50, },
4289 <pmn>                I = {20,-50},
4290 <cmr|ptm>            J = {100, },
4291 <pad>                J = {50, },
4292 <pmn>                J = {20, },
4293 <cmr|pad|ppl|ptm>    K = {50, },
4294 <pmn>                K = {20, },
4295 <cmr|pad|ppl|ptm>    L = {50, },
4296 <pmn>                L = {20,50},
4297 <ugm>                L = { ,100},
4298 <cmr|ptm>            M = {50, },
4299 <pmn>                M = { , -30},
4300 <cmr|ptm>            N = {50, },
4301 <pmn>                N = { , -30},
4302 <bch|pmn|ppl|ptm>    O = {50, },
4303 <cmr|pad>            O = {100, },
4304 <ugm>                O = {70,50},
4305 <pmn|ppl|ptm>        \OE = {50, },
4306 <pad>                \OE = {100, },
4307 <cmr|pad|ppl|ptm>    P = {50, },
4308 <pmn>                P = {20,-50},
4309 <bch|pmn|ppl|ptm>    Q = {50, },
4310 <cmr|pad>            Q = {100, },
4311 <ugm>                Q = {70,50},
4312 <cmr|pad|ppl|ptm>    R = {50, },
4313 <pmn>                R = {20, },
4314 <bch|cmr|pad|ppl|ptm> S = {50, },
4315 <pmn>                S = {20,-30},
4316 <bch|cmr|pad|ppl|ptm> $ = {50, },
4317 <pmn>                $ = {20,-30},
4318 <bch|pmn|ugm>        T = {70, },
4319 <cmr|pad|ppl|ptm>    T = {100, },
4320 <cmr|pad|ppl|ptm>    U = {50, },
4321 <pmn>                U = {50,-50},
4322 <cmr|pad|pmn|ugm>    V = {100, },
4323 <ppl|ptm>            V = {100,50},
4324 <cmr|pad|pmn|ugm>    W = {100, },
4325 <ppl>                W = {50, },
4326 <ptm>                W = {100,50},
4327 <cmr|ppl|ptm>        X = {50, },
4328 <cmr|ptm>            Y = {100, },
4329 <pmn>                Y = {50, },
4330 <ppl>                Y = {100,50},
4331 <pmn>                Z = { , -50},
4332 <pmn>                d = { , -50},
4333 <pad|pmn>            f = { , -100},
4334 <pmn>                i = { , -30},
4335 <pmn>                j = { , -30},
4336 <pmn>                l = { , -100},
4337 <bch>                o = {50,50},
4338 <bch>                p = { , 50},
4339 <pmn>                p = {-50, },
4340 <bch>                q = {50, },
4341 <pmn>                r = { , 50},

```

```

4342 <bch>      t = { ,50},
4343 <pmn|ugm>   v = {50, },
4344 <bch>      w = { ,50},
4345 <pmn|ugm>   w = {50, },
4346 <bch>      y = { ,50},
4347 <cmr>      0 = {100, },
4348 <bch|ptm>   1 = {150,100},
4349 <cmr>      1 = {200,50},
4350 <pad>      1 = {150, },
4351 <pmn>      1 = {50, },
4352 <ppl>      1 = {100, },
4353 <ugm>      1 = {150,150},
4354 <cmr>      2 = {100,-100},
4355 <pad|ppl|ptm> 2 = {50, },
4356 <pmn>      2 = {-50, },
4357 <bch>      3 = {50, },
4358 <cmr>      3 = {100,-100},
4359 <pmn>      3 = {-100, },
4360 <ptm>      3 = {100,50},
4361 <bch>      4 = {100, },
4362 <cmr|pad>   4 = {150, },
4363 <ppl|ptm>   4 = {50, },
4364 <cmr>      5 = {100, },
4365 <ptm>      5 = {50, },
4366 <bch>      6 = {50, },
4367 <cmr>      6 = {100, },
4368 <bch|pad|ptm> 7 = {100, },
4369 <cmr>      7 = {200,-150},
4370 <pmn>      7 = {20, },
4371 <ppl>      7 = {50, },
4372 <cmr>      8 = {50,-50},
4373 <cmr>      9 = {100,-100},
4374 <m-t|cmr|pad|pmn|ppl> . = { ,500},
4375 <bch|ptm|ugm> . = { ,700},
4376 <m-t|cmr|pad|pmn|ppl> {,}= { ,500},
4377 <bch|ugm>   {,}= { ,600},
4378 <ptm>      {,}= { ,700},
4379 <m-t|cmr|pad|ppl> : = { ,300},
4380 <bch|ugm>   : = { ,400},
4381 <pmn>      : = { ,200},
4382 <ptm>      : = { ,500},
4383 <m-t|cmr|pad|ppl> ; = { ,300},
4384 <bch|ugm>   ; = { ,400},
4385 <pmn>      ; = { ,200},
4386 <ptm>      ; = { ,500},
4387 <ptm>      ! = { ,100},
4388 <bch>      ? = { ,200},
4389 <ptm>      ? = { ,100},
4390 <ppl>      ? = { ,300},
4391 <pmn>      " = {400,200},
4392 <m-t|pad|pmn|ppl|ptm> & = {50,50},
4393 <bch>      & = { ,80},
4394 <cmr>      & = {100,50},
4395 <ugm>      & = {50,100},
4396 <m-t|cmr|pad|pmn> \% = {100, },
4397 <bch>      \% = {50,50},
4398 <ppl|ptm>   \% = {100,100},
4399 <ugm>      \% = {100,50},
4400 <m-t|pmn|ppl> * = {200,200},
4401 <bch>      * = {300,200},
4402 <cmr>      * = {400,100},
4403 <pad>      * = {500,100},
4404 <ptm|ugm>   * = {400,200},

```



```

4405 <m-t|cmr|pmn|ppl>      + = {150,200},
4406 <bch|ugm>               + = {250,250},
4407 <pad|ptm>               + = {250,200},
4408 <m-t|pad|pmn|ppl>      @ = {50,50},
4409 <bch>                    @ = {80,50},
4410 <cmr>                    @ = {200,50},
4411 <ptm>                    @ = {150,150},
4412 <m-t|bch|ugm>          ~ = {150,150},
4413 <cmr|pad|pmn|ppl|ptm>  ~ = {200,150},
4414 <ugm>                    {=} = {200,200},
4415 ( = {200, }, ) = { ,200},
4416 <m-t|cmr|pad|ppl|ptm|ugm> / = {100,200},
4417 <bch>                    / = { ,150},
4418 <pmn>                    / = {100,150},
4419 <m-t>                    - = {300,300},
4420 <bch|pad>                - = {300,400},
4421 <pmn>                    - = {200,300},
4422 <cmr>                    - = {500,300},
4423 <ppl>                    - = {300,500},
4424 <ptm>                    - = {500,500},
4425 <ugm>                    - = {400,700},
4426 <m-t|pmn>               \textendash = {200,200}, \textemdash = {150,150},
4427 <bch>                   \textendash = {200,300}, \textemdash = {150,200},
4428 <cmr>                   \textendash = {500,300}, \textemdash = {400,200},
4429 <pad|ppl|ptm|ugm>      \textendash = {300,300}, \textemdash = {200,200},
4430 <m-t|bch|pmn|ugm>       \textquoteleft = {400,200}, \textquoteright = {400,200},
4431 <cmr|pad>               \textquoteleft = {800,200}, \textquoteright = {800,200},
4432 <ppl>                   \textquoteleft = {700,400}, \textquoteright = {700,400},
4433 <ptm>                   \textquoteleft = {800,500}, \textquoteright = {800,500},
4434 <m-t|bch|pmn>          \textquotedblleft = {400,200}, \textquotedblright = {400,200},
4435 <cmr>                   \textquotedblleft = {700,100}, \textquotedblright = {500,300},
4436 <pad>                   \textquotedblleft = {700,200}, \textquotedblright = {700,200},
4437 <ppl>                   \textquotedblleft = {500,300}, \textquotedblright = {500,300},
4438 <ptm>                   \textquotedblleft = {700,400}, \textquotedblright = {700,400},
4439 <ugm>                   \textquotedblleft = {600,200}, \textquotedblright = {600,200},
4440 }
4441
4442 <*cmr>
4443 \SetProtrusion
4444 [ name = cmr-it-OT1,
4445   load = cmr-it ]
4446 { encoding = {OT1,OT4},
4447   family = cmr,
4448   shape = it }
4449 {
4450   \AE = {100, },
4451   \OE = {100, },
4452   "00 = {200,150}, % \Gamma
4453   "01 = {150,100}, % \Delta
4454   "02 = {150, 50}, % \Theta
4455   "03 = {150, 50}, % \Lambda
4456   "04 = {100,100}, % \Xi
4457   "05 = {100,100}, % \Pi
4458   "06 = {100, 50}, % \Sigma
4459   "07 = {200,150}, % \Upsilon
4460   "08 = {150, 50}, % \Phi
4461   "09 = {150,100}, % \Psi
4462   "0A = { 50, 50}, % \Omega
4463 }
4464
4465 </cmr>
4466 \SetProtrusion
4467 <m-t> [ name = T1-it-default,

```

```

4468 <bch> [ name      = bch-it-T1,
4469 <cmr> [ name      = cmr-it-T1,
4470 <pad> [ name      = pad-it-T1,
4471 <pmn> [ name      = pmnj-it-T1,
4472 <ppl> [ name      = ppl-it-T1,
4473 <ptm> [ name      = ptm-it-T1,
4474 <ugm> [ name      = ugm-it-T1,
4475 <m-t> load      = OT1-it ]
4476 <bch> load      = bch-it ]
4477 <cmr> load      = cmr-it ]
4478 <pmn> load      = pmnj-it ]
4479 <pad> load      = pad-it ]
4480 <ppl> load      = ppl-it ]
4481 <ptm> load      = ptm-it ]
4482 <ugm> load      = ugm-it ]
4483 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4484 <ptm|ugm> { encoding = {T1},
4485 <bch> family = bch,
4486 <cmr> family = cmr,
4487 <pmn> family = pmnj,
4488 <pad> family = {pad,padx,padj},
4489 <ppl> family = {ppl,pplx,pplj},
4490 <ptm> family = {ptm,ptmx,ptmj},
4491 <ugm> family = ugm,
4492 <!(cmr|ugm)> shape = {it,sl} }
4493 <cmr|ugm> shape = it }
4494 {
4495 <m-t|bch|pmn> _ = { ,100},
4496 <cmr|ugm> _ = {100,200},
4497 <pad|ppl|ptm> _ = {100,100},
4498 <cmr> \AE = {100, },
4499 <bch> \OE = { 50, },
4500 <cmr> \OE = {100, },
4501 <pmn> 031 = { , -100}, % ffl
4502 <cmr|ptm> 156 = {100, }, % IJ
4503 <pad> 156 = {50, }, % IJ
4504 <pmn> 156 = {20, }, % IJ
4505 <pmn> 188 = { , -30}, % ij
4506 <pmn> \v t = { ,100},
4507 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
4508 <cmr|ugm> \textbackslash = {300,300},
4509 <bch> \textbackslash = {150,150},
4510 <pmn> \textbackslash = {100,150},
4511 <ugm> \textbar = {200,200},
4512 <cmr> \textquotedblleft = {500,300},
4513 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
4514 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4515 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4516 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
4517 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
4518 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4519 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4520 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4521 <pad> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
4522 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4523 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
4524 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
4525 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4526 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
4527 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
4528 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4529 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
4530 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },

```

```

4531 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
4532 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
4533 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
4534 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
4535 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
4536 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100},
4537 <pmn> \textvisiblespace = {100,100},
4538 }
4539
4540 <*m-t|ptm>
4541 \SetProtrusion
4542 <m-t> [ name = QX-it-default,
4543 <ptm> [ name = ptm-it-QX,
4544 <m-t> load = OT1-it ]
4545 <ptm> load = ptm-it ]
4546 { encoding = {QX},
4547 <ptm> family = {ptm,ptmx,ptmj},
4548 shape = {it,sl} }
4549 {
4550 <ptm> 009 = { , 50}, % fk
4551 {=} = {100,100},
4552 <m-t> \textunderscore = {100,100},
4553 <ptm> \textunderscore = {100,150},
4554 \textbackslash = {100,200},
4555 \quotedblbase = {300,400},
4556 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
4557 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
4558 \textexclamdown = {200, }, \textquestiondown = {200, },
4559 \textbraceleft = {200,100}, \textbraceright = {200,200},
4560 \textless = {100,100}, \textgreater = {100,100},
4561 \textminus = {200,200}, \textdegree = {300,150},
4562 <m-t> \copyright = {100,100}, \textregistered = {100,100},
4563 <ptm> \textregistered = {100,150}, \copyright = {100,150},
4564 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
4565 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
4566 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
4567 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
4568 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
4569 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
4570 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
4571 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
4572 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
4573 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
4574 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
4575 <ptm> \textperthousand = { ,50},
4576 }
4577
4578 </m-t|ptm>
4579 <*cmr|bch>
4580 \SetProtrusion
4581 <cmr> [ name = cmr-it-T5,
4582 <cmr> load = cmr-it ]
4583 <bch> [ name = bch-it-T5,
4584 <bch> load = bch-it ]
4585 { encoding = T5,
4586 <bch> family = bch,
4587 <cmr> family = cmr,
4588 shape = it }
4589 {
4590 <bch> _ = { ,100},
4591 <cmr> _ = {100,200},
4592 <bch> \textbackslash = {150,150},
4593 <cmr> \textbackslash = {300,300},

```

```

4594 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4595 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4596 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4597 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4598 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
4599 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4600 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
4601 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
4602 <bch> \textless = {100, }, \textgreater = { ,100},
4603 <cmr> \textless = {300,100}, \textgreater = {200,100},
4604 }
4605
4606 </cmr|bch>

```

Slanted is very similar to italic.

```

4607 <*cmr>
4608 \SetProtrusion
4609 [ name = cmr-sl,
4610 load = cmr-it-OT1 ]
4611 { encoding = {OT1,OT4},
4612 family = cmr,
4613 shape = sl }
4614 {
4615 L = { ,50},
4616 f = { ,-50},
4617 - = {300, },
4618 \textendash = {400, }, \textemdash = {300, },
4619 }
4620
4621 \SetProtrusion
4622 [ name = cmr-sl-T1,
4623 load = cmr-it-T1 ]
4624 { encoding = {T1,LY1},
4625 family = cmr,
4626 shape = sl }
4627 {
4628 L = { ,50},
4629 f = { ,-50},
4630 - = {300, },
4631 \textendash = {400, }, \textemdash = {300, },
4632 }
4633
4634 \SetProtrusion
4635 [ name = cmr-sl-T5,
4636 load = cmr-it-T5 ]
4637 { encoding = T5,
4638 family = cmr,
4639 shape = sl }
4640 {
4641 L = { ,50},
4642 f = { ,-50},
4643 - = {300, },
4644 \textendash = {400, }, \textemdash = {300, },
4645 }
4646
4647 \SetProtrusion
4648 [ name = lmr-it-T1,
4649 load = cmr-it-T1 ]
4650 { encoding = {T1,LY1},
4651 family = lmr,
4652 shape = {it,sl} }
4653 {
4654 \textquotedblleft = {700,100},

```

```

4655     \quotedblbase      = {600,300},
4656   }
4657

```

Oldstyle numerals are slightly different.

```

4658 \SetProtrusion
4659 [ name = cmr(oldstyle)-it,
4660   load = cmr-it-T1 ]
4661 { encoding = T1,
4662   family   = {hfor,cmor},
4663   shape     = {it,sl} }
4664 {
4665   1 = {250, 50},
4666   2 = {150,-100},
4667   3 = {100,-50},
4668   4 = {150,150},
4669   6 = {200,   },
4670   7 = {200, 50},
4671   8 = {150,-50},
4672   9 = {100, 50},
4673 }
4674
4675 </cmr>
4676 <*pmn>
4677 \SetProtrusion
4678 [ name      = pmnx-it,
4679   load      = pmnj-it ]
4680 { encoding = OT1,
4681   family   = pmnx,
4682   shape     = {it,sl} }
4683 {
4684   1 = {100,150},
4685 }
4686
4687 \SetProtrusion
4688 [ name      = pmnx-it-T1,
4689   load      = pmnj-it-T1 ]
4690 { encoding = {T1,LY1},
4691   family   = pmnx,
4692   shape     = {it,sl} }
4693 {
4694   1 = {100,150},
4695 }
4696
4697 </pmn>
4698 <*ptm>
4699 \SetProtrusion
4700 [ name      = ptm-it-LY1,
4701   load      = ptm-it-T1 ]
4702 { encoding = {LY1},
4703   family   = {ptm,ptmx,ptmj},
4704   shape     = {it,sl} }
4705 {
4706   -                      = {100,100},
4707   \texttrademark         = {100,100},
4708   \textregistered       = {100,100},
4709   \textcopyright         = {100,100},
4710   \textdegree           = {300,100},
4711   \textminus            = {200,200},
4712   \textellipsis         = {100,200},
4713   \texteuro             = {   ,   },
4714   \textcent             = {100,100},
4715   \textquotesingle      = {500,   },

```

```

4716 \textflorin           = {100, 70},
4717 \textdagger           = {150,150},
4718 \textdaggerdbl        = {100,100},
4719 \textbullet           = {150,150},
4720 \textonesuperior      = {150,100},
4721 \texttwosuperior      = {150, 50},
4722 \textthreesuperior    = {150, 50},
4723 \textparagraph        = {100,  },
4724 \textperiodcentered    = {500,300},
4725 \textonequarter       = { 50,  },
4726 \textonehalf         = { 50,  },
4727 \textplusminus        = {100,100},
4728 \textmultiply         = {150,150},
4729 \textdivide          = {150,150},
4730 }
4731
4732 </ptm>

```

15.7.3 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).

```

4733 <!*ugm>
4734 \SetProtrusion
4735 <m-t> [ name      = OT1-sc,
4736 <bch> [ name      = bch-sc,
4737 <cmr> [ name      = cmr-sc-OT1,
4738 <pad> [ name      = pad-sc,
4739 <pmn> [ name      = pmnj-sc,
4740 <ppl> [ name      = ppl-sc,
4741 <ptm> [ name      = ptm-sc,
4742 <m-t> load      = default ]
4743 <bch> load      = bch-default ]
4744 <cmr> load      = cmr-OT1 ]
4745 <pad> load      = pad-default ]
4746 <pmn> load      = pmnj-default ]
4747 <ppl> load      = ppl-default ]
4748 <ptm> load      = ptm-default ]
4749 <m-t|bch|pad|pmn> { encoding = OT1,
4750 <cmr|ppl|ptm> { encoding = {OT1,OT4},
4751 <bch> family     = bch,
4752 <cmr> family     = cmr,
4753 <pad> family     = {pad,padx,padj},
4754 <pmn> family     = pmnj,
4755 <ppl> family     = {ppl,pplx,pplj},
4756 <ptm> family     = {ptm,ptmx,ptmj},
4757 shape          = sc }
4758 {
4759 a = {50,50},
4760 <cmr|pad|ppl|ptm> \ae = {50,  },
4761 <bch|pmn> c = {50,  },
4762 <bch|pad|pmn> d = { ,50},
4763 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
4764 <bch|pad|pmn> g = {50,  },
4765 <m-t|cmr|pad|pmn|ppl|ptm> j = {50,  },
4766 <bch> j = {100,  },
4767 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
4768 <ptm> l = { ,80},
4769 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
4770 <ptm> 013 = { ,80}, % fl

```

```

4771 <bch|pad|pmn>      o = {50,50},
4772 <pad|pmn>      \oe = {50,  },
4773 <ppl>      p = { 0, 0},
4774 <bch|pad|pmn>      q = {50,70},
4775 <ppl>      q = { 0,  },
4776 <m-t|cmr|pad|pmn|ppl|ptm>      r = {  , 0},
4777      t = {50,50},
4778 <m-t|bch|cmr|pad|pmn|ppl>      y = {50,50},
4779 <ptm>      y = {80,80},
4780      }
4781
4782 \SetProtrusion
4783 <m-t>      [ name      = Tl-sc,
4784 <bch>      [ name      = bch-sc-Tl,
4785 <cmr>      [ name      = cmr-sc-Tl,
4786 <pad>      [ name      = pad-sc-Tl,
4787 <pmn>      [ name      = pmnj-sc-Tl,
4788 <ppl>      [ name      = ppl-sc-Tl,
4789 <ptm>      [ name      = ptm-sc-Tl,
4790 <m-t>      load      = Tl-default ]
4791 <bch>      load      = bch-Tl      ]
4792 <cmr>      load      = cmr-Tl      ]
4793 <pad>      load      = pad-Tl      ]
4794 <pmn>      load      = pmnj-Tl     ]
4795 <ppl>      load      = ppl-Tl      ]
4796 <ptm>      load      = ptm-Tl      ]
4797      { encoding = {Tl,Ly1},
4798 <bch>      family   = bch,
4799 <cmr>      family   = cmr,
4800 <pad>      family   = {pad,padx,padj},
4801 <pmn>      family   = pmnj,
4802 <ppl>      family   = {ppl,pplx,pplj},
4803 <ptm>      family   = {ptm,ptmx,ptmj},
4804      shape      = sc }
4805      {
4806      a = {50,50},
4807 <cmr|pad|ppl|ptm>      \ae = {50,  },
4808 <bch|pmn>      c = {50,  },
4809 <bch|pad|pmn>      d = {  ,50},
4810 <m-t|bch|cmr|pad|pmn|ptm>      f = {  ,50},
4811 <bch|pad|pmn>      g = {50,  },
4812 <m-t|cmr|pad|pmn|ppl|ptm>      j = {50,  },
4813 <bch>      j = {100,  },
4814 <m-t|bch|cmr|pad|pmn|ppl>      l = {  ,50},
4815 <ptm>      l = {  ,80},
4816 <m-t|bch|cmr|pad|pmn|ppl>      029 = {  ,50}, % fl
4817 <ptm>      029 = {  ,80}, % fl
4818 <bch|pad|pmn>      o = {50,50},
4819 <bch|pad|pmn>      \oe = {50,  },
4820 <ppl>      p = { 0, 0},
4821 <bch|pad|pmn>      q = {50,70},
4822 <ppl>      q = { 0,  },
4823 <m-t|cmr|pad|pmn|ppl|ptm>      r = {  , 0},
4824      t = {50,50},
4825 <m-t|bch|cmr|pad|pmn|ppl>      y = {50,50},
4826 <ptm>      y = {80,80},
4827      }
4828
4829 <!!ugm>
4830 <*m-t>
4831 \SetProtrusion
4832      [ name      = QX-sc,
4833      load      = QX-default ]

```

```

4834 { encoding = QX,
4835     shape   = sc }
4836 {
4837     a = {50,50},
4838     f = { ,50},
4839     j = {50, },
4840     l = { ,50},
4841     013 = { ,50}, % f1
4842     r = { , 0},
4843     t = {50,50},
4844     y = {50,50},
4845 }
4846
4847 </m-t>
4848 <*cmr|bch>
4849 \SetProtrusion
4850 <bch> [ name      = bch-sc-T5,
4851 <bch>    load      = bch-T5 ]
4852 <cmr> [ name      = cmr-sc-T5,
4853 <cmr>    load      = cmr-T5 ]
4854 { encoding = T5,
4855 <bch>    family    = bch,
4856 <cmr>    family    = cmr,
4857     shape    = sc }
4858 {
4859     a = {50,50},
4860 <bch>    c = {50, },
4861 <bch>    d = { ,50},
4862     f = { ,50},
4863 <bch>    g = {50, },
4864 <bch>    j = {100, },
4865 <cmr>    j = {50, },
4866     l = { ,50},
4867 <bch>    o = {50,50},
4868 <bch>    q = { 0, },
4869 <cmr>    r = { , 0},
4870     t = {50,50},
4871     y = {50,50},
4872 }
4873
4874 </cmr|bch>
4875 <*pmn>
4876 \SetProtrusion
4877 [ name      = pmnx-sc,
4878   load      = pmnj-sc ]
4879 { encoding = OT1,
4880   family    = pmnx,
4881   shape     = sc }
4882 {
4883     1 = {230,180},
4884 }
4885
4886 \SetProtrusion
4887 [ name      = pmnx-sc-T1,
4888   load      = pmnj-sc-T1 ]
4889 { encoding = {T1,LY1},
4890   family    = pmnx,
4891   shape     = sc }
4892 {
4893     1 = {230,180},
4894 }
4895

```


15.7.4 Italic Small Caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's `fontinstallationguide` suggests `si`.

```

4896 \SetProtrusion
4897 [ name      = pmnj-scit,
4898   load      = pmnj-it  ]
4899 { encoding = OT1,
4900   family   = pmnj,
4901   shape     = {scit,si} }
4902 {
4903   a = {50, },
4904   \ae = { , -50},
4905   b = {20, -50},
4906   c = {50, -50},
4907   d = {20, 0},
4908   e = {20, -50},
4909   f = {10, 0},
4910   012 = {10, -50}, % fi
4911   013 = {10, -50}, % fl
4912   014 = {10, -50}, % ffi
4913   015 = {10, -50}, % ffl
4914   g = {50, -50},
4915   i = {20, -50},
4916   j = {20, 0},
4917   k = {20, },
4918   l = {20, 50},
4919   m = { , -30},
4920   n = { , -30},
4921   o = {50, },
4922   \oe = {50, -50},
4923   p = {20, -50},
4924   q = {50, },
4925   r = {20, 0},
4926   s = {20, -30},
4927   t = {70, },
4928   u = {50, -50},
4929   v = {100, },
4930   w = {100, },
4931   y = {50, },
4932   z = { , -50},
4933 }
4934
4935 \SetProtrusion
4936 [ name      = pmnj-scit-T1,
4937   load      = pmnj-it-T1  ]
4938 { encoding = {T1,LY1},
4939   family   = pmnj,
4940   shape     = {scit,si}   }
4941 {
4942   a = {50, },
4943   \ae = { , -50},
4944   b = {20, -50},
4945   c = {50, -50},
4946   d = {20, 0},
4947   e = {20, -50},
4948   f = {10, 0},
4949   028 = {10, -50}, % fi
4950   029 = {10, -50}, % fl
4951   030 = {10, -50}, % ffi
4952   031 = {10, -50}, % ffl
4953   g = {50, -50},

```

```

4954     i = {20,-50},
4955 188 = {20, 0}, % ij
4956     j = {20, 0},
4957     k = {20,  },
4958     l = {20,50},
4959     m = {  ,-30},
4960     n = {  ,-30},
4961     o = {50,  },
4962 \oe = {50,-50},
4963     p = {20,-50},
4964     q = {50,  },
4965     r = {20, 0},
4966     s = {20,-30},
4967     t = {70,  },
4968     u = {50,-50},
4969     v = {100,  },
4970     w = {100,  },
4971     y = {50,  },
4972     z = {  ,-50},
4973   }
4974
4975 \SetProtrusion
4976 [ name      = pmnx-scit,
4977   load      = pmnj-scit ]
4978 { encoding = OT1,
4979   family   = pmnx,
4980   shape    = {scit,si} }
4981 {
4982   1 = {100,150},
4983 }
4984
4985 \SetProtrusion
4986 [ name      = pmnx-scit-T1,
4987   load      = pmnj-scit-T1 ]
4988 { encoding = {T1,LY1},
4989   family   = pmnx,
4990   shape    = {scit,si}   }
4991 {
4992   1 = {100,150},
4993 }
4994
4995 </pmn>

```

15.7.5 textcomp

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

```

4996 \SetProtrusion
4997 <m-t> [ name      = textcomp ]
4998 <bch> [ name      = bch-textcomp ]
4999 <cmr> [ name      = cmr-textcomp ]
5000 <pad> [ name      = pad-textcomp ]
5001 <pmn> [ name      = pmn-textcomp ]
5002 <ppl> [ name      = ppl-textcomp ]
5003 <ptm> [ name      = ptm-textcomp ]
5004 <ugm> [ name      = ugm-textcomp ]
5005 <m-t> { encoding = TS1      }
5006 <!m-t> { encoding = TS1,
5007 <bch>   family   = bch }
5008 <cmr>   family   = cmr }
5009 <pad>   family   = {pad,padx,padj} }

```

```

5010 <pmn> family = {pmnx,pmnj} }
5011 <ppl> family = {ppl,pplx,pplj} }
5012 <ptm> family = {ptm,ptmx,ptmj} }
5013 <ugm> family = ugm }
5014 {
5015 <cmr> \textquotestraightbase = {300,300},
5016 <pad|pmn> \textquotestraightbase = {400,400},
5017 <cmr|pmn> \textquotestraightdblbase = {300,300},
5018 <pad> \textquotestraightdblbase = {400,400},
5019 <bch|cmr|pad|pmn|ugm> \texttwelvewardash = {200,200},
5020 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
5021 <ugm> \textthreequartersemdash = {200,200},
5022 <cmr|pmn> \textquotesingle = {300,400},
5023 <pad> \textquotesingle = {400,500},
5024 <ptm> \textquotesingle = {500,500},
5025 <ugm> \textquotesingle = {300,500},
5026 <bch|cmr|pmn> \textasteriskcentered = {200,300},
5027 <pad> \textasteriskcentered = {300,300},
5028 <ugm> \textasteriskcentered = {100,200},
5029 <pmn> \textfractionsolidus = {-200,-200},
5030 <cmr> \textoneoldstyle = {100,100},
5031 <pmn> \textoneoldstyle = { , 50},
5032 <cmr> \textthreeoldstyle = { , 50},
5033 <pad|pmn> \textthreeoldstyle = { 50, },
5034 <cmr> \textfouroldstyle = { 50, 50},
5035 <pad|pmn> \textfouroldstyle = { 50, },
5036 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
5037 <cmr> \textlangle = {400, },
5038 <cmr> \textrangle = { , 400},
5039 <m-t|bch|pmn|ptm> \textminus = {200,200},
5040 <cmr|pad|ppl> \textminus = {300,300},
5041 <ugm> \textminus = {250,300},
5042 <bch|pad|pmn> \textlbrackdbl = {100, },
5043 <bch|pad|pmn> \textrbrackdbl = { , 100},
5044 <pmn> \textasciigrave = {200,500},
5045 <bch|cmr|pad|pmn> \texttildelow = {200,250},
5046 <pmn> \textasciibreve = {300,400},
5047 <pmn> \textasciicaron = {300,400},
5048 <pmn> \textacutedbl = {200,300},
5049 <pmn> \textgravedbl = {150,300},
5050 <bch|pmn|ugm> \textdagger = { 80, 80},
5051 <cmr|pad> \textdagger = {100,100},
5052 <ptm> \textdagger = {150,150},
5053 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
5054 <ptm> \textdaggerdbl = {100,100},
5055 <bch> \textbardbl = {100,100},
5056 <ugm> \textbardbl = {150,150},
5057 <bch> \textbullet = {200,200},
5058 <cmr|pad|pmn> \textbullet = { , 100},
5059 <ptm> \textbullet = {150,150},
5060 <ugm> \textbullet = { 50, 100},
5061 <bch|cmr|pmn> \textcelsius = { 50, },
5062 <pad> \textcelsius = { 80, },
5063 <bch> \textflorin = { 50, 50},
5064 <pad|ugm> \textflorin = { , 100},
5065 <pmn> \textflorin = { 50, 100},
5066 <ptm> \textflorin = { 50, 70},
5067 <cmr> \textcolonmonetary = { , 50},
5068 <pad|pmn> \textcolonmonetary = { 50, },
5069 <pmn> \textinterrobang = { , 100},
5070 <pmn> \textinterrobangdown = {100, },
5071 <m-t|pad|ptm> \texttrademark = {100,100},
5072 <bch> \texttrademark = {150,150},

```

```

5073 <cmr|ppl> \texttrademark = {200,200},
5074 <pmn> \texttrademark = { 50, 50},
5075 <ugm> \texttrademark = {100,150},
5076 <bch|ugm> \textcent = { 50, },
5077 <ptm> \textcent = {100,100},
5078 <bch> \textsterling = { 50, },
5079 <ugm> \textsterling = { , 50},
5080 <bch> \textbrokenbar = {200,200},
5081 <ugm> \textbrokenbar = {200,300},
5082 <pmn> \textasciidieresis = {300,400},
5083 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
5084 <pmn> \textcopyright = {100,150},
5085 <ppl> \textcopyright = {200,200},
5086 <bch|cmr|ugm> \textordfeminine = {100,200},
5087 <pad|pmn> \textordfeminine = {200,200},
5088 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
5089 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
5090 <pmn> \textregistered = { 50,150},
5091 <ppl> \textregistered = {200,200},
5092 <pmn> \textasciimacron = {150,200},
5093 <m-t|ppl|ptm> \textdegree = {300,300},
5094 <bch> \textdegree = {150,200},
5095 <cmr|pad> \textdegree = {400,400},
5096 <pmn> \textdegree = {150,400},
5097 <ugm> \textdegree = {200,200},
5098 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
5099 <ptm> \textpm = { 50, 80},
5100 <bch|ugm> \texttwosuperior = {100,200},
5101 <cmr> \texttwosuperior = { 50,100},
5102 <pad|pmn> \texttwosuperior = {200,200},
5103 <ptm> \texttwosuperior = { 50, 50},
5104 <bch|ugm> \textthreesuperior = {100,200},
5105 <cmr> \textthreesuperior = { 50,100},
5106 <pad|pmn> \textthreesuperior = {200,200},
5107 <ptm> \textthreesuperior = { 50, 50},
5108 <pmn> \textasciicute = {300,400},
5109 <bch|ugm> \textmu = { ,100},
5110 <bch|pad|pmn> \textparagraph = { ,100},
5111 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
5112 <ptm> \textperiodcentered = {300,300},
5113 <ugm> \textperiodcentered = {200,500},
5114 <bch|ugm> \textonesuperior = {200,300},
5115 <cmr|pad|pmn> \textonesuperior = {200,200},
5116 <ptm> \textonesuperior = {100,100},
5117 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
5118 <cmr> \textordmasculine = {100,200},
5119 <bch|cmr|pmn> \texteuro = {100, },
5120 <pad> \texteuro = { 50,100},
5121 <bch|ptm> \texttimes = {100,100},
5122 <cmr> \texttimes = {150,250},
5123 <pad> \texttimes = {100,150},
5124 <pmn> \texttimes = { 70,100},
5125 <ugm> \texttimes = {200,300},
5126 <bch|pad|pmn> \textdiv = {150,200},
5127 <cmr> \textdiv = {150,250},
5128 <ptm> \textdiv = { 50,100},
5129 <ugm> \textdiv = {200,300},
5130 <ptm> \textperthousand = { ,50},
5131 <ugm> \textsection = { ,100},
5132 <ugm> \textonehalf = { 50,100},
5133 <ugm> \textonequarter = { 50,100},
5134 <ugm> \textthreequarters = { 50,100},
5135 <ugm> \textsurd = { ,100},

```

Remaining slots in the source file.

```

5136 }
5137
5138 <*cmr|pad|pmn|ugm>
5139 \SetProtrusion
5140 <cmr> [ name = cmr-textcomp-it ]
5141 <pad> [ name = pad-textcomp-it ]
5142 <pmn> [ name = pmn-textcomp-it ]
5143 <ugm> [ name = ugm-textcomp-it ]
5144 { encoding = TS1,
5145 <cmr> family = cmr,
5146 <pad> family = {pad,padx,padj},
5147 <pmn> family = {pmnx,pmnj},
5148 <ugm> family = ugm,
5149 <!ugm> shape = {it,sl} }
5150 <ugm> shape = it }
5151 {
5152 <cmr> \textquotestraightbase = {300,600},
5153 <pad|pmn> \textquotestraightbase = {400,400},
5154 <cmr> \textquotestraightdblbase = {300,600},
5155 <pad> \textquotestraightdblbase = {300,400},
5156 <pmn> \textquotestraightdblbase = {300,300},
5157 \texttwelveudash = {200,200},
5158 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
5159 <ugm> \textthreequartersemdash = {200,200},
5160 <cmr> \textquotesingle = {600,300},
5161 <pad> \textquotesingle = {800,100},
5162 <pmn> \textquotesingle = {300,200},
5163 <ugm> \textquotesingle = {500,500},
5164 <cmr> \textasteriskcentered = {300,200},
5165 <pad> \textasteriskcentered = {500,100},
5166 <pmn> \textasteriskcentered = {200,300},
5167 <ugm> \textasteriskcentered = {300,150},
5168 <pmn> \textfractionsolidus = {-200,-200},
5169 <cmr> \textoneoldstyle = {100, 50},
5170 <pad> \textoneoldstyle = {100, },
5171 <pmn> \textoneoldstyle = { 50, },
5172 <pad> \texttwooldstyle = { 50, },
5173 <pmn> \texttwooldstyle = {-50, },
5174 <cmr> \textthreeoldstyle = {100, 50},
5175 <pmn> \textthreeoldstyle = {-100, },
5176 <cmr> \textfouroldstyle = { 50, 50},
5177 <pad> \textfouroldstyle = { 50,100},
5178 <cmr> \textsevenoldstyle = { 50, 80},
5179 <pad> \textsevenoldstyle = { 50, },
5180 <pmn> \textsevenoldstyle = { 20, },
5181 <cmr> \textlangle = {400, },
5182 <cmr> \textrangle = { ,400},
5183 <cmr|pad> \textminus = {300,300},
5184 <pmn> \textminus = {200,200},
5185 <ugm> \textminus = {250,300},
5186 <pad|pmn> \textlbrackdbl = {100, },
5187 <pad|pmn> \textrbrackdbl = { ,100},
5188 <pmn> \textasciigrave = {300,300},
5189 <cmr|pad|pmn> \texttildelow = {200,250},
5190 <pmn> \textasciibreve = {300,300},
5191 <pmn> \textasciicaron = {300,300},
5192 <pmn> \textacutedbl = {200,300},
5193 <pmn> \textgravedbl = {150,300},
5194 <cmr> \textdagger = {100,100},
5195 <pad> \textdagger = {200,100},
5196 <pmn> \textdagger = { 80, 50},

```

```

5197 <ugm> \textdagger = { 80, 80},
5198 <cmr|pad> \textdaggerdbl = { 80, 80},
5199 <pmn> \textdaggerdbl = { 80, 50},
5200 <ugm> \textbardbl = {150,150},
5201 <cmr> \textbullet = {200,100},
5202 <pad> \textbullet = {300, },
5203 <pmn> \textbullet = { 30, 70},
5204 <ugm> \textbullet = { 50,100},
5205 <cmr> \textcelsius = {100, },
5206 <pad> \textcelsius = {200, },
5207 <pmn> \textcelsius = { 50,-50},
5208 <pad> \textflorin = {100, },
5209 <pmn> \textflorin = { 50,100},
5210 <ugm> \textflorin = { ,100},
5211 <cmr> \textcolonmonetary = {150, },
5212 <pad> \textcolonmonetary = {100, },
5213 <pmn> \textcolonmonetary = { 50,-50},
5214 <cmr|pad> \texttrademark = {200, },
5215 <pmn> \texttrademark = { 50,100},
5216 <ugm> \texttrademark = {150, 50},
5217 <ugm> \textcent = { 50, },
5218 <ugm> \textsterling = { , 50},
5219 <ugm> \textbrokenbar = {200,300},
5220 <pmn> \textasciidieresis = {300,200},
5221 <cmr> \textcopyright = {100, },
5222 <pad> \textcopyright = {200,100},
5223 <pmn> \textcopyright = {100,150},
5224 <ugm> \textcopyright = {300, },
5225 <cmr> \textordfeminine = {100,100},
5226 <pmn> \textordfeminine = {200,200},
5227 <ugm> \textordfeminine = {100,200},
5228 <cmr|pad> \textlnot = {300, },
5229 <pmn|ugm> \textlnot = {200, },
5230 <cmr> \textregistered = {100, },
5231 <pad> \textregistered = {200,100},
5232 <pmn> \textregistered = { 50,150},
5233 <ugm> \textregistered = {300, },
5234 <pmn> \textasciimacron = {150,200},
5235 <cmr|pad> \textdegree = {500,100},
5236 <pmn> \textdegree = {150,150},
5237 <ugm> \textdegree = {300,200},
5238 <cmr> \textpm = {150,100},
5239 <pad> \textpm = {200,150},
5240 <pmn|ugm> \textpm = {150,200},
5241 <cmr> \textonesuperior = {400, },
5242 <pad> \textonesuperior = {300,100},
5243 <pmn> \textonesuperior = {200,100},
5244 <ugm> \textonesuperior = {300,300},
5245 <cmr> \texttwosuperior = {400, },
5246 <pad> \texttwosuperior = {300, },
5247 <pmn> \texttwosuperior = {200,100},
5248 <ugm> \texttwosuperior = {300,200},
5249 <cmr> \textthreesuperior = {400, },
5250 <pad> \textthreesuperior = {300, },
5251 <pmn> \textthreesuperior = {200,100},
5252 <ugm> \textthreesuperior = {300,200},
5253 <ugm> \textmu = { ,100},
5254 <pmn> \textasciacute = {300,200},
5255 <cmr> \textparagraph = {200, },
5256 <pmn> \textparagraph = { ,100},
5257 <cmr> \textperiodcentered = {500,500},
5258 <pad|pmn|ugm> \textperiodcentered = {300,400},
5259 <cmr> \textordmasculine = {100,100},

```

```

5260 <pmn> \textordmasculine = {200,200},
5261 <ugm> \textordmasculine = {300,200},
5262 <cmr> \texteuro = {200, },
5263 <pad> \texteuro = {100, },
5264 <pmn> \texteuro = {100,-50},
5265 <cmr> \texttimes = {200,200},
5266 <pad> \texttimes = {200,100},
5267 <pmn> \texttimes = { 70,100},
5268 <ugm> \texttimes = {200,300},
5269 <cmr|pad> \textdiv = {200,200},
5270 <pmn> \textdiv = {150,200},
5271 <ugm> \textdiv = {200,300},
5272 <ugm> \textsection = { ,200},
5273 <ugm> \textonehalf = { 50,100},
5274 <ugm> \textonequarter = { 50,100},
5275 <ugm> \textthreequarters = { 50,100},
5276 <ugm> \textsurd = { ,100},
5277 }
5278
5279 </cmr|pad|pmn|ugm>

```

15.7.6 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}

```

```

5280 <*cmr>
5281 \SetProtrusion
5282 [ name = cmr-math-letters ]
5283 { encoding = OML,
5284   family = cmm,
5285   series = {m,b},
5286   shape = it }
5287 {
5288   A = {100, 50}, % \mathnormal
5289   B = { 50, },
5290   C = { 50, },
5291   D = { 50, 50},
5292   E = { 50, },
5293   F = {100, 50},
5294   G = { 50, 50},
5295   H = { 50, 50},
5296   I = { 50, 50},
5297   J = {150, 50},
5298   K = { 50,100},
5299   L = { 50, 50},
5300   M = { 50, },

```

```

5301     N = { 50,   },
5302     O = { 50,   },
5303     P = { 50,   },
5304     Q = { 50, 50},
5305     R = { 50,   },
5306     S = { 50,   },
5307     T = { 50,100},
5308     U = { 50, 50},
5309     V = {100,100},
5310     W = { 50,100},
5311     X = { 50,100},
5312     Y = {100,100},
5313     f = {100,100},
5314     h = {   ,100},
5315     i = {   , 50},
5316     j = {   , 50},
5317     k = {   , 50},
5318     r = {   , 50},
5319     v = {   , 50},
5320     w = {   , 50},
5321     x = {   , 50},
5322     "0B = { 50,100}, % \alpha
5323     "0C = { 50, 50}, % \beta
5324     "0D = {200,150}, % \gamma
5325     "0E = { 50, 50}, % \delta
5326     "0F = { 50, 50}, % \epsilon
5327     "10 = { 50,150}, % \zeta
5328     "12 = { 50,   }, % \theta
5329     "13 = {   ,100}, % \iota
5330     "14 = {   ,100}, % \kappa
5331     "15 = {100, 50}, % \lambda
5332     "16 = {   , 50}, % \mu
5333     "17 = {   , 50}, % \nu
5334     "18 = {   , 50}, % \xi
5335     "19 = { 50,100}, % \pi
5336     "1A = { 50, 50}, % \rho
5337     "1B = {   ,150}, % \sigma
5338     "1C = { 50,150}, % \tau
5339     "1D = { 50, 50}, % \upsilon
5340     "1F = { 50,100}, % \chi
5341     "20 = { 50, 50}, % \psi
5342     "21 = {   , 50}, % \omega
5343     "22 = {   , 50}, % \varepsilon
5344     "23 = {   , 50}, % \vartheta
5345     "24 = {   , 50}, % \varpi
5346     "25 = {100,   }, % \varrho
5347     "26 = {100,100}, % \varsigma
5348     "27 = { 50, 50}, % \varphi
5349     "28 = {100,100}, % \leftharpoonup
5350     "29 = {100,100}, % \leftharpoondown
5351     "2A = {100,100}, % \rightharpoonup
5352     "2B = {100,100}, % \rightharpoondown
5353     "2C = {300,200}, % \lhook
5354     "2D = {200,300}, % \rhook
5355     "2E = {   ,100}, % \triangleright
5356     "2F = {100,   }, % \triangleleft
5357     "3A = {   ,500}, % ., \ldotp
5358     "3B = {   ,500}, % ,
5359     "3C = {200,100}, % <
5360     "3D = {300,400}, % /
5361     "3E = {100,200}, % >
5362     "3F = {200,200}, % \star
5363     "5B = {   ,100}, % \flat

```



```

5364 "5E = {200,200}, % \smile
5365 "5F = {200,200}, % \frown
5366 "7C = {100, }, % \jmath
5367 "7D = { ,100}, % \wp

```

Remaining slots in the source file.

```

5368 }
5369

```

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}

```

```

5370 \SetProtrusion
5371 [ name = cmr-math-symbols ]
5372 { encoding = OMS,
5373   family = cmsy,
5374   series = {m,b},
5375   shape = n }
5376 {
5377   A = {150, 50}, % \mathcal
5378   C = { ,100},
5379   D = { , 50},
5380   F = { 50,150},
5381   I = { ,100},
5382   J = {100,150},
5383   K = { ,100},
5384   L = {100, },
5385   M = { 50, 50},
5386   N = { 50,100},
5387   P = { , 50},
5388   Q = { 50, },
5389   R = { , 50},
5390   T = { 50,150},
5391   V = { 50, 50},
5392   W = { , 50},
5393   X = {100,100},
5394   Y = {100, },
5395   Z = {100,150},
5396   "00 = {300,300}, % -
5397   "01 = { ,700}, % \cdot, \dotp
5398   "02 = {150,250}, % \times
5399   "03 = {150,250}, % *, \ast
5400   "04 = {200,300}, % \div
5401   "05 = {150,250}, % \diamond
5402   "06 = {200,200}, % \pm
5403   "07 = {200,200}, % \mp
5404   "08 = {100,100}, % \oplus
5405   "09 = {100,100}, % \ominus
5406   "0A = {100,100}, % \otimes
5407   "0B = {100,100}, % \oslash
5408   "0C = {100,100}, % \odot
5409   "0D = {100,100}, % \bigcirc
5410   "0E = {100,100}, % \circ
5411   "0F = {100,100}, % \bullet
5412   "10 = {100,100}, % \asymp
5413   "11 = {100,100}, % \equiv
5414   "12 = {200,100}, % \subseteq
5415   "13 = {100,200}, % \supseteq
5416   "14 = {200,100}, % \leq
5417   "15 = {100,200}, % \geq
5418   "16 = {200,100}, % \preceq

```

```

5419 "17 = {100,200}, % \succeq
5420 "18 = {200,200}, % \sim
5421 "19 = {150,150}, % \approx
5422 "1A = {200,100}, % \subset
5423 "1B = {100,200}, % \supset
5424 "1C = {200,100}, % \ll
5425 "1D = {100,200}, % \gg
5426 "1E = {300,100}, % \prec
5427 "1F = {100,300}, % \succ
5428 "20 = {100,200}, % \leftarrow
5429 "21 = {200,100}, % \rightarrow
5430 "22 = {100,100}, % \uparrow
5431 "23 = {100,100}, % \downarrow
5432 "24 = {100,100}, % \leftrightarrow
5433 "25 = {100,100}, % \nearrow
5434 "26 = {100,100}, % \searrow
5435 "27 = {100,100}, % \simeq
5436 "28 = {100,100}, % \Leftarrow
5437 "29 = {100,100}, % \Rightarrow
5438 "2A = {100,100}, % \Uparrow
5439 "2B = {100,100}, % \Downarrow
5440 "2C = {100,100}, % \Leftrightarrow
5441 "2D = {100,100}, % \nrightarrow
5442 "2E = {100,100}, % \swarrow
5443 "2F = { ,100}, % \propto
5444 "30 = { ,400}, % \prime
5445 "31 = {100,100}, % \infty
5446 "32 = {150,100}, % \in
5447 "33 = {100,150}, % \ni
5448 "34 = {100,100}, % \triangle, \bigtriangleup
5449 "35 = {100,100}, % \bigtriangledown
5450 "38 = { ,100}, % \forall
5451 "39 = {100, }, % \exists
5452 "3A = {200, }, % \neg
5453 "3E = {200,200}, % \top
5454 "3F = {200,200}, % \bot, \perp
5455 "5E = {100,200}, % \wedge
5456 "5F = {100,200}, % \vee
5457 "60 = { ,300}, % \vdash
5458 "61 = {300, }, % \dashv
5459 "62 = {100,100}, % \lfloor
5460 "63 = {100,100}, % \rfloor
5461 "64 = {100,100}, % \lceil
5462 "65 = {100,100}, % \rceil
5463 "66 = {150, }, % \lbrace
5464 "67 = { ,150}, % \rbrace
5465 "68 = {400, }, % \langle
5466 "69 = { ,400}, % \rangle
5467 "6C = {100,100}, % \updownarrow
5468 "6D = {100,100}, % \Updownarrow
5469 "6E = {100,300}, % \, \backslash, \setminus
5470 "72 = {100,100}, % \nabla
5471 "79 = {200,200}, % \dagger
5472 "7A = {100,100}, % \ddagger
5473 "7B = {100, }, % \mathparagraph
5474 "7C = {100,100}, % \clubsuit
5475 "7D = {100,100}, % \diamondsuit
5476 "7E = {100,100}, % \heartsuit
5477 "7F = {100,100}, % \spadesuit

```

Remaining slots in the source file.

```

5478 }
5479

```

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

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

```
5480 </cmr>
5481 </cfg-t>
```

15.7.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
5482 <*cfg-u>
```

Symbol font 'a'.

```
5483 <*msa>
5484 \SetProtrusion
5485   [ name      = AMS-a ]
5486   { encoding = U,
5487     family   = msa }
5488   {
5489     "05 = {150,250}, % \centerdot
5490     "06 = {100,100}, % \lozenge
5491     "07 = { 50, 50}, % \blacklozenge
5492     "08 = { 50, 50}, % \circlearrowright
5493     "09 = { 50, 50}, % \circlearrowleft
5494     "0A = {100,100}, % \rightleftharpoons
5495     "0B = {100,100}, % \leftrightharpoons
5496     "0D = {-50,200}, % \Vdash
5497     "0E = {-50,200}, % \Vvdash
5498     "0F = {-70,150}, % \vDash
5499     "10 = {100,150}, % \twoheadrightarrow
5500     "11 = {100,150}, % \twoheadleftarrow
5501     "12 = { 50,100}, % \leftleftarrows
5502     "13 = { 50, 80}, % \rightrightarrows
5503     "14 = {120,120}, % \upuparrows
5504     "15 = {120,120}, % \downdownarrows
5505     "16 = {200,200}, % \upharpoonright
5506     "17 = {200,200}, % \downharpoonright
5507     "18 = {200,200}, % \upharpoonleft
5508     "19 = {200,200}, % \downharpoonleft
5509     "1A = { 80,100}, % \rightarrowtail
5510     "1B = { 80,100}, % \leftarrowtail
5511     "1C = { 50, 50}, % \leftrightarrows
5512     "1D = { 50, 50}, % \rightleftarrows
5513     "1E = {250,  }, % \Lsh
5514     "1F = {  ,250}, % \Rsh
5515     "20 = {100,100}, % \rightsquigarrow
5516     "21 = {100,100}, % \leftrightsquigarrow
5517     "22 = {100, 50}, % \looparrowleft
5518     "23 = { 50,100}, % \looparrowright
5519     "24 = { 50, 80}, % \circeq
5520     "25 = {  ,100}, % \succsim
5521     "26 = {  ,100}, % \gtrsim
5522     "27 = {  ,100}, % \gtrapprox
5523     "28 = {150, 50}, % \multimap
5524     "2B = {100,150}, % \doteqdot
5525     "2C = {100,150}, % \triangleq
5526     "2D = {100, 50}, % \precsim
5527     "2E = {100, 50}, % \lessim
5528     "2F = { 50, 50}, % \lessapprox
5529     "30 = {100, 50}, % \eqslantless
```

```

5530 "31 = { 50, 50}, % \eqslantgtr
5531 "32 = {100, 50}, % \curlyeqprec
5532 "33 = { 50,100}, % \curlyeqsucc
5533 "34 = {100, 50}, % \preccurlyeq
5534 "36 = { 50,  }, % \leqslant
5535 "38 = {  , 50}, % \backprime
5536 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
5537 "3C = { 50,100}, % \succcurlyeq
5538 "3E = {  , 50}, % \geqslant
5539 "40 = {  , 50}, % \sqsubset
5540 "41 = { 50,  }, % \sqsupset
5541 "42 = {  ,150}, % \vartriangleright, \rhd
5542 "43 = {150,  }, % \vartriangleleft, \lhd
5543 "44 = {  ,100}, % \trianglerighteq, \unrhd
5544 "45 = {100,  }, % \trianglelefteq, \unlhd
5545 "46 = {100,100}, % \bigstar
5546 "48 = { 50, 50}, % \blacktriangledown
5547 "49 = {  ,100}, % \blacktriangleright
5548 "4A = {100,  }, % \blacktriangleleft
5549 "4B = {  ,150}, % \dashrightarrow (the arrow)
5550 "4C = {150,  }, % \dashleftarrow
5551 "4D = { 50, 50}, % \vartriangle
5552 "4E = { 50, 50}, % \blacktriangle
5553 "4F = { 50, 50}, % \triangledown
5554 "50 = { 50, 50}, % \eqcirc
5555 "56 = {  ,150}, % \rightarrow
5556 "57 = {150,  }, % \leftarrow
5557 "58 = {100,300}, % \checkmark
5558 "5C = { 50, 50}, % \angle
5559 "5D = { 50, 50}, % \measuredangle
5560 "5E = { 50, 50}, % \sphericalangle
5561 "5F = {  , 50}, % \varpropto
5562 "60 = {100,100}, % \smallsmile
5563 "61 = {100,100}, % \smallfrown
5564 "62 = { 50,  }, % \Subset
5565 "63 = {  , 50}, % \Supset
5566 "66 = {150,150}, % \curlywedge
5567 "67 = {150,150}, % \curlyvee
5568 "68 = { 50,150}, % \leftthreetimes
5569 "69 = {100, 50}, % \rightthreetimes
5570 "6C = { 50, 50}, % \bumpeq
5571 "6D = { 50, 50}, % \Bumpeq
5572 "6E = {100,  }, % \lll
5573 "6F = {  ,100}, % \ggg
5574 "70 = { 50,100}, % \ulcorner
5575 "71 = {100, 50}, % \urcorner
5576 "75 = {150,200}, % \dotplus
5577 "76 = { 50,100}, % \backsim
5578 "78 = { 50,100}, % \llcorner
5579 "79 = {100, 50}, % \lrcorner
5580 "7C = {100,100}, % \intercal
5581 "7D = { 50, 50}, % \circledcirc
5582 "7E = { 50, 50}, % \circledast
5583 "7F = { 50, 50}, % \circleddash

```

Remaining slots in the source file.

```

5584 }
5585
5586 </msa>

```

Symbol font 'b'.

```

5587 <*msb>
5588 \SetProtrusion

```

```

5589 [ name      = AMS-b ]
5590 { encoding = U,
5591   family   = msb }
5592 {
5593   A = { 50, 50}, % \mathbb
5594   C = { 50, 50},
5595   G = {   , 50},
5596   L = {   , 50},
5597   P = {   , 50},
5598   R = {   , 50},
5599   T = {   , 50},
5600   V = { 50, 50},
5601   X = { 50, 50},
5602   Y = { 50, 50},
5603   "00 = { 50, 50}, % \lvertneqq
5604   "01 = { 50, 50}, % \gvertneqq
5605   "02 = { 50, 50}, % \nleq
5606   "03 = { 50, 50}, % \ngeq
5607   "04 = {100, 50}, % \nless
5608   "05 = { 50,150}, % \ngtr
5609   "06 = {100, 50}, % \nprec
5610   "07 = { 50,150}, % \nsucc
5611   "08 = { 50, 50}, % \lneqq
5612   "09 = { 50, 50}, % \gneqq
5613   "0A = {100,100}, % \nleqslant
5614   "0B = {100,100}, % \ngeqslant
5615   "0C = {100, 50}, % \lneq
5616   "0D = { 50,100}, % \gneq
5617   "0E = {100, 50}, % \npreceq
5618   "0F = { 50,100}, % \nsucceq
5619   "10 = { 50,   }, % \precnsim
5620   "11 = { 50, 50}, % \succnsim
5621   "12 = { 50, 50}, % \lnsim
5622   "13 = { 50, 50}, % \gnsim
5623   "14 = { 50, 50}, % \nleqq
5624   "15 = { 50, 50}, % \ngeqq
5625   "16 = { 50, 50}, % \precneqq
5626   "17 = { 50, 50}, % \succneqq
5627   "18 = { 50, 50}, % \precnapprox
5628   "19 = { 50, 50}, % \succnapprox
5629   "1A = { 50, 50}, % \lnapprox
5630   "1B = { 50, 50}, % \gnapprox
5631   "1C = {150,200}, % \nsim
5632   "1D = { 50, 50}, % \ncong
5633   "1E = {100,150}, % \diagup
5634   "1F = {100,150}, % \diagdown
5635   "20 = {100, 50}, % \varsubsetneq
5636   "21 = { 50,100}, % \varsupsetneq
5637   "22 = {100, 50}, % \nsubseteqq
5638   "23 = { 50,100}, % \nsupseteqq
5639   "24 = {100, 50}, % \subsetneqq
5640   "25 = { 50,100}, % \supsetneqq
5641   "26 = {100, 50}, % \varsubsetneqq
5642   "27 = { 50,100}, % \varsupsetneqq
5643   "28 = {100, 50}, % \subsetneq
5644   "29 = { 50,100}, % \supsetneq
5645   "2A = {100, 50}, % \nsubseteq
5646   "2B = { 50,100}, % \nsupseteq
5647   "2C = { 50,100}, % \nparallel
5648   "2D = {100,150}, % \nmid
5649   "2E = {150,150}, % \nshortmid
5650   "2F = {100,100}, % \nshortparallel
5651   "30 = {   ,150}, % \nvdash

```

```

5652 "31 = { ,150}, % \nVdash
5653 "32 = { ,100}, % \nvDash
5654 "33 = { ,100}, % \nVDash
5655 "34 = { ,100}, % \ntrianglerighteq
5656 "35 = {100, }, % \ntrianglelefteq
5657 "36 = {100, }, % \ntriangleleft
5658 "37 = { ,100}, % \ntriangleright
5659 "38 = {100,200}, % \nleftarrow
5660 "39 = {100,200}, % \nrightharpoonleft
5661 "3A = {100,100}, % \nLeftarrow
5662 "3B = { 50,100}, % \nRrightarrow
5663 "3C = {100,100}, % \nLeftrightarrow
5664 "3D = {100,200}, % \nleftrightharpoonleft
5665 "3E = { 50, 50}, % \divideontimes
5666 "3F = { 50, 50}, % \varnothing
5667 "60 = {200, }, % \Finv
5668 "61 = { , 50}, % \Game
5669 "68 = {100,100}, % \eqsim
5670 "69 = { 50, }, % \beth
5671 "6A = { 50, }, % \gimel
5672 "6B = {150, }, % \daleth
5673 "6C = {200, }, % \lessdot
5674 "6D = { ,200}, % \gtrdot
5675 "6E = {100,200}, % \ltimes
5676 "6F = {150,100}, % \rtimes
5677 "70 = { 50,100}, % \shortmid
5678 "71 = { 50, 50}, % \shortparallel
5679 "72 = {200,300}, % \smallsetminus
5680 "73 = {100,200}, % \thicksim
5681 "74 = { 50,100}, % \thickapprox
5682 "75 = { 50, 50}, % \approxeq
5683 "76 = { 50,100}, % \succapprox
5684 "77 = { 50, 50}, % \precapprox
5685 "78 = {100,100}, % \curvearrowleft
5686 "79 = { 50,150}, % \curvearrowright
5687 "7A = { 50,200}, % \digamma
5688 "7B = {100, 50}, % \varkappa
5689 "7F = {200, }, % \backepsilon

```

Remaining slots in the source file.

```

5690 }
5691
5692 </msb>

```

15.7.8 Euler

Euler Roman font (package eulr).

```

5693 <eur>
5694 \SetProtrusion
5695 [ name = eulr ]
5696 { encoding = U,
5697   family = eur }
5698 {
5699   "01 = {100,100},
5700   "03 = {100,150},
5701   "06 = { ,100},
5702   "07 = {100,150},
5703   "08 = {100,100},
5704   "0A = {100,100},
5705   "0B = { , 50},
5706   "0C = { ,100},

```

```

5707 "0D = {100,100},
5708 "0E = { ,100},
5709 "0F = {100,100},
5710 "10 = {100,100},
5711 "13 = { ,100},
5712 "14 = { ,100},
5713 "15 = { , 50},
5714 "16 = { , 50},
5715 "17 = { 50,100},
5716 "18 = { 50,100},
5717 "1A = { , 50},
5718 "1B = { , 50},
5719 "1C = { 50,100},
5720 "1D = { 50,100},
5721 "1E = { 50,100},
5722 "1F = { 50,100},
5723 "20 = { , 50},
5724 "21 = { , 50},
5725 "22 = { 50,100},
5726 "24 = { , 50},
5727 "27 = { 50,100},
5728 "1 = {100,100},
5729 "7 = { 50,100},
5730 "3A = {300,500},
5731 "3B = {200,400},
5732 "3C = {200,100},
5733 "3D = {200,200},
5734 "3E = {100,200},
5735 "A = { ,100},
5736 "D = { , 50},
5737 "J = { 50, },
5738 "K = { , 50},
5739 "L = { , 50},
5740 "Q = { , 50},
5741 "T = { 50, },
5742 "X = { 50, 50},
5743 "Y = { 50, },
5744 "h = { , 50},
5745 "k = { , 50},
5746 }
5747

```

Extended by the `eulervm` package.

```

5748 \SetProtrusion
5749 [ name      = euler-vm,
5750   load      = euler ]
5751 { encoding = U,
5752   family   = zeur }
5753 {
5754   "28 = {100,200},
5755   "29 = {100,200},
5756   "2A = {100,150},
5757   "2B = {100,150},
5758   "2C = {200,300},
5759   "2D = {200,300},
5760   "2E = { ,100},
5761   "2F = {100, },
5762   "3F = {150,150},
5763   "5B = { ,100},
5764   "5E = {100,100},
5765   "5F = {100,100},
5766   "80 = { , 50},
5767   "81 = {200,250},

```

```

5768     "82 = {100,200},
5769   }
5770
5771 </eur>

```

Euler Script font (euca1).

```

5772 <*eus>
5773 \SetProtrusion
5774 [ name      = euscript ]
5775 { encoding = U,
5776   family   = eus  }
5777 {
5778   A = {100,100},
5779   B = { 50,100},
5780   C = { 50, 50},
5781   D = { 50,100},
5782   E = { 50,100},
5783   F = { 50,  },
5784   G = { 50,  },
5785   H = {  ,100},
5786   K = {  , 50},
5787   L = {  ,150},
5788   M = {  , 50},
5789   N = {  , 50},
5790   O = { 50, 50},
5791   P = { 50, 50},
5792   T = {  ,100},
5793   U = {  , 50},
5794   V = { 50, 50},
5795   W = { 50, 50},
5796   X = { 50, 50},
5797   Y = { 50,  },
5798   Z = { 50,100},
5799   "00 = {250,250},
5800   "18 = {200,200},
5801   "3A = {200,150},
5802   "40 = {  ,100},
5803   "5E = {100,100},
5804   "5F = {100,100},
5805   "66 = { 50,  },
5806   "67 = {  , 50},
5807   "6E = {200,200},
5808 }
5809
5810 \SetProtrusion
5811 [ name      = euscript-vm,
5812   load      = euscript ]
5813 { encoding = U,
5814   family   = zeus  }
5815 {
5816   "01 = {600,600},
5817   "02 = {200,200},
5818   "03 = {200,200},
5819   "04 = {200,200},
5820   "05 = {150,150},
5821   "06 = {200,200},
5822   "07 = {200,200},
5823   "08 = {100,100},
5824   "09 = {100,100},
5825   "0A = {100,100},
5826   "0B = {100,100},
5827   "0C = {100,100},
5828   "0D = {100,100},

```



```
5829 "0E = {150,150},
5830 "0F = {100,100},
5831 "10 = {150,150},
5832 "11 = {100,100},
5833 "12 = {150,100},
5834 "13 = {100,150},
5835 "14 = {150,100},
5836 "15 = {100,150},
5837 "16 = {200,100},
5838 "17 = {100,200},
5839 "19 = {150,150},
5840 "1A = {150,100},
5841 "1B = {100,150},
5842 "1C = {100,100},
5843 "1D = {100,100},
5844 "1E = {250,100},
5845 "1F = {100,250},
5846 "20 = {150,200},
5847 "21 = {150,200},
5848 "22 = {150,150},
5849 "23 = {150,150},
5850 "24 = {100,200},
5851 "25 = {150,150},
5852 "26 = {150,150},
5853 "27 = {100,100},
5854 "28 = {100,100},
5855 "29 = {100,150},
5856 "2A = {100,100},
5857 "2B = {100,100},
5858 "2C = {100,100},
5859 "2D = {150,150},
5860 "2E = {150,150},
5861 "2F = {100,100},
5862 "30 = {100,100},
5863 "31 = {100,100},
5864 "32 = {100,100},
5865 "33 = {100,100},
5866 "34 = {100,100},
5867 "35 = {100,100},
5868 "3E = {150,150},
5869 "3F = {150,150},
5870 "60 = { ,200},
5871 "61 = {200, },
5872 "62 = {100,100},
5873 "63 = {100,100},
5874 "64 = {100,100},
5875 "65 = {100,100},
5876 "68 = {300, },
5877 "69 = { ,300},
5878 "6C = {100,100},
5879 "6D = {100,100},
5880 "6F = {100,100},
5881 "72 = {100,100},
5882 "73 = {200,100},
5883 "76 = { ,100},
5884 "77 = {100, },
5885 "78 = { 50, 50},
5886 "79 = {100,100},
5887 "7A = {100,100},
5888 "7D = {150,150},
5889 "7E = {100,100},
5890 "A8 = {100,100},
5891 "A9 = {100,100},
```

```

5892     "AB  = {200,200},
5893     "BA  = {   ,200},
5894     "BB  = {   ,200},
5895     "BD  = {200,200},
5896     "DE  = {200,200},
5897   }
5898
5899 </eus>

```

Euler Fraktur font (eufrak).

```

5900 <*euf>
5901 \SetProtrusion
5902   [ name      = mathfrak ]
5903   { encoding = U,
5904     family   = euf   }
5905   {
5906     A  = {   , 50},
5907     B  = {   , 50},
5908     C  = { 50, 50},
5909     D  = {   , 80},
5910     E  = { 50,   },
5911     G  = {   , 50},
5912     L  = {   , 80},
5913     O  = {   , 50},
5914     T  = {   , 80},
5915     X  = { 80, 50},
5916     Z  = { 80, 50},
5917     b  = {   , 50},
5918     c  = {   , 50},
5919     k  = {   , 50},
5920     p  = {   , 50},
5921     q  = { 50,   },
5922     v  = {   , 50},
5923     w  = {   , 50},
5924     x  = {   , 50},
5925     1  = {100,100},
5926     2  = { 80, 80},
5927     3  = { 80, 50},
5928     4  = { 80, 50},
5929     7  = { 50, 50},
5930     "12 = {500,500},
5931     "13 = {500,500},
5932     !  = {   ,200},
5933     '  = {200,300},
5934     (  = {200,   },
5935     )  = {   ,200},
5936     *  = {200,200},
5937     +  = {200,250},
5938     -  = {200,200},
5939     {,} = {300,300},
5940     .  = {400,400},
5941     {=} = {200,200},
5942     :  = {   ,200},
5943     ;  = {   ,200},
5944     ]  = {   ,200},
5945   }
5946
5947 </euf>
5948 </cfg-u>

```

15.7.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym¹⁴).

```

5949 <*cfg-e>
5950 \SetProtrusion
5951 <zpeu|euroitc> { encoding = U,
5952 <mvs> { encoding = {OT1,U},
5953 <zpeu> family = zpeu }
5954 <euroitc> family = {euroitc,euroitcs} }
5955 <mvs> family = mvs }
5956 {
5957 <zpeu> E = {50, }
5958 <euroitc> E = {100,50}
5959 <mvs> 164 = {50,50}, % \EUR
5960 <mvs> 068 = {50,-100}, % \EURdig
5961 }
5962
5963 <*zpeu|euroitc>
5964 \SetProtrusion
5965 { encoding = U,
5966 <zpeu> family = zpeu,
5967 <euroitc> family = {euroitc,euroitcs},
5968 shape = it* }
5969 {
5970 <zpeu> E = {100,-50}
5971 <euroitc> E = {100,}
5972 }
5973
5974 </zpeu|euroitc>
5975 <*zpeu>
5976 \SetProtrusion
5977 { encoding = U,
5978 family = {zpeus,eurosans} }
5979 {
5980 E = {100,50}
5981 }
5982
5983 \SetProtrusion
5984 { encoding = U,
5985 family = {zpeus,eurosans},
5986 shape = it* }
5987 {
5988 E = {200, }
5989 }
5990
5991 </zpeu>
5992 </cfg-e>

```

15.8 Interword Spacing

Default unit is space.

```

5993 <*m-t>
5994 %%% -----
5995 %%% INTERWORD SPACING SETTINGS
5996
5997 \SetExtraSpacing
5998 [ name = default ]

```

¹⁴ Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

```
5999 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6000 {
```

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

```
6001 {,} = { , -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)

```
6002 r = { , -300, 300},
```

- before or after lowercase characters with ascenders

```
6003 b = { , -200, 200},
6004 d = { , -200, 200},
6005 f = { , -200, 200},
6006 h = { , -200, 200},
6007 k = { , -200, 200},
6008 l = { , -200, 200},
6009 t = { , -200, 200},
```

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

```
6010 c = { , -100, 100},
6011 p = { , -100, 100},
6012 v = { , -100, 100},
6013 w = { , -100, 100},
6014 z = { , -100, 100},
6015 x = { , -100, 100},
6016 y = { , -100, 100}, % ?
```

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

```
6017 i = { , 50, -50},
6018 m = { , 50, -50},
6019 n = { , 50, -50},
6020 u = { , 50, -50},
```

- after colon and semicolon

```
6021 : = { , 200, -200},
6022 ; = { , 200, -200},
```

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

```

6023      . = { ,250,-250},
6024      ! = { ,250,-250},
6025      ? = { ,250,-250},

```

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

```

6026      }
6027

```

Questions are:

- Is the result really better?
- Is it overdone? (Try with a factor < 1000.)
- Should the first parameter also be used? (Probably.)

15.8.1 Nonfrenchspacing

The following settings simulate `\nonfrenchspacing` (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the `TEXbook`:

‘If the space factor f is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [...] Then the stretch component is multiplied by $f / 1000$, while the shrink component is multiplied by $1000 / f$.’

The ‘extra space’ (`\fontdimen 7`) for Computer Modern Roman is a third of `\fontdimen 2`, i. e., 333.

```

6028 \SetExtraSpacing
6029 [ name      = nonfrench-cmr,
6030   load       = default,
6031   context    = nonfrench ]
6032 { encoding = {OT1,T1,LV1,OT4,QX,T5},
6033   family    = cmr }
6034 {

```

`latex.ltx` has:

```

\def\nonfrenchspacing{
  \sfcode`\.\ 3000

```

```

6035      . = {333,2000,-667},

```

```

\sfcode`\? 3000

```

```

6036      ? = {333,2000,-667},

```

```

\sfcode`\! 3000

```

```

6037      ! = {333,2000,-667},

```

```

\sfcode`\: 2000

```

```

6038      : = {333,1000,-500},

```

```

\sfcode`\; 1500

```

```
6039      ; = {      , 500,-333},
```

```
\sfcode`\, 1250
```

```
6040      {,}= {      , 250,-200},
```

```
}
```

```
6041 }
```

```
6042
```

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.

```
6043 \SetExtraSpacing
6044 [ name      = nonfrench-default,
6045   load      = default,
6046   context   = nonfrench ]
6047 { encoding = {OT1,T1,LV1,OT4,QX,T5} }
6048 {
6049   . = {240,2000,-667},
6050   ? = {240,2000,-667},
6051   ! = {240,2000,-667},
6052   : = {240,1000,-500},
6053   ; = {      , 500,-333},
6054   {,}= {      , 250,-200},
6055 }
6056
```

15.9 Additional Kerning

Default unit is 1em.

```
6057 %%% -----
6058 %%% ADDITIONAL KERNING
6059
```

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

```
6060 \SetExtraKerning
6061 [ name = empty ]
6062 { encoding = {OT1,T1,LV1,OT4,QX,T5,TS1} }
6063 { }
6064
```

15.9.1 French

For the French context.

```
6065 \SetExtraKerning
6066 [ name      = french-default,
6067   context   = french,
6068   unit      = space ]
6069 { encoding = {OT1,T1,LV1} }
6070 {
6071   : = {1000,}, % = \fontdimen2
6072   ; = {500, }, % ~ \thinspace
6073   ! = {500, },
6074   ? = {500, },
6075 }
6076
```

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

```

6077 \SetExtraKerning
6078   [ name      = french-guillemets,
6079     context    = french-guillemets,
6080     load       = french-default,
6081     unit       = space ]
6082   { encoding = {T1,LY1} }
6083   {
6084     \guillemotleft = { ,800}, % = 0.8\fontdimen2
6085     \guillemotright = {800, },
6086   }
6087
6088 \SetExtraKerning
6089   [ name      = french-guillemets-OT1,
6090     context    = french-guillemets,
6091     load       = french-default,
6092     unit       = space ]
6093   { encoding = OT1 }
6094   { }
6095
```

15.9.2 Turkish

And for Turkish.

```

6096 \SetExtraKerning
6097   [ name      = turkish,
6098     context    = turkish,
6099     unit       = space ]
6100   { encoding = {OT1,T1,LY1} }
6101   {
6102     : = {500, }, % ~ \thinspace
6103     ! = {500, },
6104     {=} = {500, },
6105   }
6106
6107 </m-t>
6108 </config>

```

16 Auxiliary File for Micro Fine Tuning

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

```

6109 <*test>
6110 \documentclass{article}
6111
6112 %% Here you can specify the font you want to test, using
6113 %% the commands \fontfamily, \fontseries and \fontshape.
6114 %% Make sure to end all lines with a comment character!
6115 \newcommand*\TestFont{%
6116   \fontfamily{ppl}%
6117   \fontseries{b}%
6118   \fontshape{it}% sc, sl
6119 }
6120
6121 \usepackage{ifthen}
6122 \usepackage[T1]{fontenc}
6123 \usepackage[latin1]{inputenc}
6124 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}

```

```

6125
6126 \pagestyle{empty}
6127 \setlength{\parindent}{0pt}
6128 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
6129 \newcommand*\testprotrusion[2][{}]{%
6130   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2%
6131   lorem ipsum dolor sit amet,
6132   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
6133   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6134   you know the rest%
6135   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill} #2%
6136   \linebreak
6137   {\fontencoding{\encodingdefault}%
6138    \fontseries{\seriesdefault}%
6139    \fontshape{\shapedefault}%
6140    \selectfont
6141    Here is the beginning of a line, \dotfill and here is its end}\linebreak
6142 }
6143 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
6144 \def\stripprefix#1>{}
6145 \newcount\charcount
6146 \begin{document}
6147
6148 \microtypesetup{expansion=false}
6149
6150 {\centering The font in this document is called by:\\
6151 \texttt{\showTestFont}\par}\bigskip
6152
6153 \TestFont\selectfont
6154 This line intentionally left empty\linebreak
6155 %% A -- Z
6156 \charcount=65
6157 \loop
6158   \testprotrusion{\char\charcount}
6159   \advance\charcount 1
6160   \ifnum\charcount < 91 \repeat
6161 %% a -- z
6162 \charcount=97
6163 \loop
6164   \testprotrusion{\char\charcount}
6165   \advance\charcount 1
6166   \ifnum\charcount < 123 \repeat
6167 %% 0 -- 9
6168 \charcount=48
6169 \loop
6170   \testprotrusion{\char\charcount}
6171   \advance\charcount 1
6172   \ifnum\charcount < 58 \repeat
6173 %%
6174 \testprotrusion[r]{,}
6175 \testprotrusion[r]{.}
6176 \testprotrusion[r]{;}
6177 \testprotrusion[r]{:}
6178 \testprotrusion[r]{?}
6179 \testprotrusion[r]{!}
6180 \testprotrusion[l]{\textexclamdown}
6181 \testprotrusion[l]{\textquestiondown}
6182 \testprotrusion[r]{\{}}
6183 \testprotrusion[l]{\{ }
6184 \testprotrusion{/}
6185 \testprotrusion{\char~}
6186 \testprotrusion{-}
6187 \testprotrusion{\textendash}

```



```
6188 \testprotrusion{\textendash}
6189 \testprotrusion{\textquoteleft}
6190 \testprotrusion{\textquoteright}
6191 \testprotrusion{\textquotedblleft}
6192 \testprotrusion{\textquotedblright}
6193 \testprotrusion{\quotesinglbase}
6194 \testprotrusion{\quotedblbase}
6195 \testprotrusion{\guilsinglleft}
6196 \testprotrusion{\guilsinglright}
6197 \testprotrusion{\guillemotleft}
6198 \testprotrusion{\guillemotright}
6199
6200 \newpage
6201 The following displays the current font stretched by 5%,
6202 normal, and shrunk by 5%:
6203
6204 \bigskip
6205 \newlength{\MTln}
6206 \newcommand*{\teststring}
6207   {ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
6208 \settowidth{\MTln}{\teststring}
6209 \microtypesetup{expansion=true}
6210
6211 \parbox{1.05\MTln}{\teststring\linebreak\
6212                  \teststring}\par\bigskip
6213 \parbox{0.95\MTln}{\teststring}
6214
6215 \end{document}
6216 /test
```

Needless to say that things may always be improved. For suggestions, mail to w.m.1@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>)	65	\MT@get@basefamily: only remove suffix, if it is 'x' or 'j'	67
issue an error instead of a warning, when pdfTeX version is too old for autoexpand	107	\MT@get@listname@: don't check for empty attributes list	67
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i>)	113	\MT@ifempty: fix: use category code 12 for the percent character (reported by <i>Tom Kink</i>)	37
Protrusion: add factors for some more characters settings for Adobe Minion (contributed by <i>Harald Harders</i>)	118	\MT@is@number: numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i>)	71
\DeclareCharacterInheritance: new command: possibility to specify character inheritance . .	91	\MT@pdfTeX@no: fix concerning version check (reported by <i>Harald Harders</i>)	32
\MT@declare@sets: remove spaces around set name	80	\MT@permute: don't use sets for empty encoding . .	93
\MT@DeclareSet@: remove spaces around first argument	79	\MT@pr@split: fix: allow zero and negative values . .	50
\MT@find@file: fix: also check whether the file for the base font family has already been loaded	65	\MT@use@set: remove spaces around set name	84
		\UseMicrotypeSet: remove spaces around first argument	83

Version 1.2 (2004/10/03)

General: check for packages that might load fonts .	75	\MT@get@highlevel: check whether defaults have changed	81
check whether only one encoding specified . . .	92	\MT@get@listname@: alternatively check for alias font name	67
Font Sets: declare cmr as an alias of cmr	112	\MT@get@size: additional magic to catch some errors hijack \set@fontsize instead of \set@fontsize	82
new: allmath and basicmath	111	\MT@get@slot: fix: group must also include \MT@get@composite	69
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding	138	\MT@loop: fix: new macro, used instead of \loop . .	40
add settings for Computer Modern Roman math symbols	143	\MT@maybe@do: also check for alias font name	47
\MT@context: fix: set inheritance list \globally to \empty	69	\MT@permute@@@@: more sanity checks for \SetProtrusion and \SetExpansion	94
\MT@familyalias: define alias font name as an alternative, not as a replacement	47	\MT@setupfont: also search for alias font file	46
\MT@get@basefamily: also remove 'w' (swash capitals)	67	fix: call \@@enc@update if necessary	46

Version 1.3 (2004/10/27)

Font Sets: declare aer, zer and hfor as an alias of cmr	112	\MT@define@opt@keys: fix: specifying load option does no longer require to give a name, too . .	89
\MT@catcodes: check some category codes (compatibility with german)	33	\MT@load@list: check whether list exists	65

Version 1.4 (2004/11/12)

General: don't use scratch registers in global definitions	69	(OT1, T1, lmr)	123
no need to check for packages that might load fonts anymore	75	\microtypesetup: fix: set the correct levels, and remember them; warning when enabling an option disabled in package options	102
use \pickup@font instead of \define@newfont as the hook for \MT@setupfont	75	\MT@pdfcprot@error: check for pdfcprot	43
use one instead of five counters	41	\SetExpansion: fix: specifying extra options does no longer require to give a name, too	86
Protrusion: tweak quote characters for cmr variants			

Version 1.4a (2004/11/17)

General: new option: final	99	codes when reading files (reported by <i>Michael Hoppe</i>)	66
\MT@normal@catcodes: fix: reset some more cat-			

Version 1.4b (2004/11/26)

General: fix: set catcodes before reading global con-		name if encoding failed	48
figuration file (reported by <i>Christoph Bier</i>)	101	\MT@get@basefamily: fix: failed for font names of the	
new message if \pdfoutput is changed	105	form abczz (reported by <i>Georg Verwey</i>) ..	67
optimisation: use less \expandafers and		\MT@get@slot: don't define \MT@char globally (save	
\csnames	36	stack problem)	69
Protrusion: harmonise dashes in upshape and italic		\MT@ifdimen: don't set \MT@count globally (save stack	
(cmr, pad, ppl)	118	problem)	38
slanted like italics	126	\MT@use@set: don't use undeclared font sets	84
\MT@checklist@family: fix: don't try alias family			

Version 1.5 (2004/12/15)

General: defaults: step: 4 (suggested by <i>Hàn Thế Thành</i>)	99	\MT@catcodes: reset catcode of ‘’ (compatibility with	
defaults: calculate step as min(stretch,shrink)/5		chemsym)	33
.....	106	\MT@get@highlevel: don't test defaults if called after	
defaults: turn off expansion for DVI output ..	105	begin document	81
disable automatic expansion for DVI output ..	106	\MT@normal@catcodes: reset catcode of ‘=’ (compati-	
new option: selected, by default false (sug-		bility with Turkish babel)	66
gested by <i>Hàn Thế Thành</i>)	98	\MT@scale@factor: warning for factors outside limits	52
Documentation: add note about DVIOutput option ..	8	\MT@scale@to@em: don't use \lpcode and \rprcode for	
add short history (section 13)	26	the calculation	51
Inheritance: remove \ss from T1 list, add \DJ ..	114	\MT@set@ex@codes: allow non-selected font expan-	
Protrusion: settings for Bitstream Charter	119	sion	56
\DeclareMicrotypeAlias: remove spaces around ar-		\MT@set@pr@codes: adjust protrusion factors before	
guments	85	setting the inheriting characters	49

Version 1.6 (2005/01/24)

General: defaults: turn off expansion for old pdf \TeX		\MT@def@num@opt: test whether numeric options re-	
versions	100	ceive a number	99
disable automatic expansion for old pdf \TeX ver-		\MT@get@charwd: use e- \TeX 's \fontcharwd, if avail-	
sions	107	able	51
load a font, if none is active	46	\MT@get@inh@list: correct message if selected is	
new option: factor, by default 1000	99	false	69
restructure dtx file	111	\MT@set@ex@codes: introduce factor option	56
test whether \pickup@font has changed	76	\MT@set@pr@codes: introduce factor option	49
use e- \TeX 's \ifcsname and \ifdefined if defined	36	\MT@use@set: retain current set if new set is unde-	
Protrusion: add italic uppercase Greek letters ...	126	clared	84
improve settings for numbers (pointed out by		\MT@vinfo: new macro: used instead of	
<i>Peter Muthesius</i>)	121	\ifMT@verbose	30
tune CMR math letters (OML encoding)	143		

Version 1.6a (2005/02/02)

Documentation: add table of fonts with tailored pro-		reported by <i>Bernard Gaulle</i>)	69
trusion settings	18	\MT@pdftex@no: new macro	32
\MT@get@slot: completely redone, hopefully more		\MT@reset@ef@codes: only reset \efcodes for older	
robust (compatible with frenchpro; problem		pdf \TeX versions	56

Version 1.7 (2005/03/23)

General: \SetExpansion: bug fix: remove space after autoexpand	91	\MT@get@slot: remove backslash hack	69
\SetExpansion: don't allow automatic expansion for old pdf \TeX versions	91	test for \chardefed commands	70
allow specification of size ranges (suggested by <i>Andreas Böhmann</i>)	81	test whether \(\encoding)\(\dots) is defined	70
modify \showhyphens	107	\MT@if@list@exists: don't define \MT@#1@name globally, here and elsewhere	68
new value for verbose option: errors	99	\MT@if@dimen: comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i>) ...	38
shorter command names	41	\MT@increment: use e- \TeX 's \numexpr if available ..	41
warning when running in draft mode	105	\MT@is@composite: new macro: construct command for composite character; no uncontrolled expansion	73
Documentation: add hint about compatibility	23	\MT@normal@catcodes: reset catcode of ':' (compatibility with french* packages)	66
remove table of match order	12	\MT@scale: new macro: use e- \TeX 's \numexpr if available	41
Protrusion: fix: remove \ from OT1, add \textbackslash to T1 encoding	122	\MT@set@ex@codes: two versions of this macro	56
\DeclareMicrotypeAlias: may also be used inside configuration files	85	\MT@split@name: don't define \MT@encoding &c. globally	47
\LoadMicrotypeFile: new command (suggested by <i>Andreas Böhmann</i>)	85	\MT@test@ast: make it simpler	81
\MicrotypeHook: new command for font package authors	101	\MT@try@order: always check for size, too (suggested by <i>Andreas Böhmann</i>)	67
\microtypesetup: fix: warning also when setting to (no)compatibility	102	fix: also check for //(\series)/(\shape)// (reported by <i>Andreas Böhmann</i>)	67
\MT@begin@catcodes: also use inside configuration commands	66	\MT@warn@code@too@large: new macro: type out maximum protrusion factor	52
\MT@get@listname@: use \tfor (<i>Andreas Böhmann</i> 's idea)	67	\MT@warn@err: new macro: for verbose=errors ...	30

Version 1.8 (2005/06/23)

General: \SetProtrusion: new key: unit	90	\MT@get@font@dimen@six: new macro: test whether \fontdimen 6 is defined	50
if font substitution has occurred, set up the substitute font, not the selected one	75	\MT@get@listname@: made recursive	67
new option: config to load a different main configuration file	100	\MT@get@slot: fix: expand active characters	70
new option: unit, by default character	100	test whether \(\encoding)\(\dots) is defined made more robust	70
Documentation: add example for factor option ..	13	\MT@get@unit: new macro: get unit for codes	54
add example of how to get rid of a widow (suggested by <i>Adam Kucharczyk</i>)	14	\MT@in@list: made recursive	40
add hint about error messages	23	\MT@is@active: new macro: translate inputenc-defined characters	71
Font Sets: add U encoding to allmath	111	\MT@is@letter: warning for non-ASCII characters ..	70
declare pxx and txr as aliases of ppl resp. ptm	112	\MT@led@kern: character protrusion with ledmac ..	43
Inheritance: remove \DJ from T1 list (it's the same as \DH)	114	\MT@make@string: use \@onelevel@sanitize	41
Protrusion: add LY1 characters for Times	125	\MT@map@clist@n: new macro: used instead of \@for	39
settings for AMS math fonts	147	\MT@map@tlist@n: new macro: used instead of \tfor	39
verified settings for slanted Computer Modern Roman	132	\MT@normal@catcodes: reset catcodes of the remaining ASCII characters	66
\DeclareMicrotypeAlias: warning when overriding an alias font	85	\MT@old@cmd: renamed commands from \..MicroType.. to \..Microtype..	30
\DeclareMicrotypeSetDefault: new command: set default font set	84	\MT@orig@add@accent: fix: disable micro-typographic setup inside \add@accent (reported by <i>Stephan Hennig</i>)	76
\MT@check@rlist: made recursive	95	\MT@pdf@tex@no: case 5: pdf \TeX 1.30	32
\MT@curr@list@name: new macro: current list type and name	73	\MT@permute@@@@@: add ranges to the beginning of the lists	94
\MT@declare@sets: warning when redefining a set ..	80	\MT@pr@split: get character width once only	50
\MT@define@set@key@: use comma lists instead of token lists	80	\MT@scale: fix: remove spaces in non-e- \TeX variant (reported by <i>Mark Rossi</i>)	41
\MT@find@file: no longer wrap names in commands	65	\MT@setupfont@hook: restore \% and \# when hyperref is loaded	44
\MT@get@charwd: warning for missing (resp. zero-width) characters	51		

restore csquotes's active characters	44	\MT@use@set: fix: remove braces in first line	84
restore percent character if Spanish babel is loaded	44	\MT@xadd: simplified	39

Version 1.9 (2005/10/28)

General: \DeclareMicrotypeSet: new key: font	82	settings for OT4 encoding (Computer Modern	
\SetProtrusion and \SetExpansion: new key:		Roman, Palatino, Times)	118
font	88	settings for T5 encoded Computer Modern Ro-	
\SetProtrusion: value ‘relative’ renamed to		man	118
‘character’ for key unit	90	\DisableLigatures: new command: disable ligatures	
allow context-specific font setup	75	(requires pdfTeX 1.30)	84
disable expansion if both step and shrink are		\microtypecontext: new command: change setup	
zero	107	context in the document	78
disable microtype setup inside hyperref’s		\MT@checklist@family: fix: add two missing	
\pdfstringdef (reported by <i>Hàn Thế Thành</i>)	44	\xexpandafters	48
option unit: rename value relative to		\MT@define@option: fix: use true as the default value	97
character	100	\MT@detokenize@c: fix the non-e-TeX version	37
warning if user requested zero step	106	\MT@exp@two@n: new macros: less \xexpandafters	36
Documentation: add hint about verbatim environ-		\MT@get@opt: new key ‘preset’ to set all characters	
ment	23	to the specified value before loading the lists	53
add remark about Type 1 fonts required for au-		\MT@is@active: redone: use \set@display@protect	71
tomatic font expansion	7	\MT@is@letter: using \catcode should be more effi-	
Font Sets: add OT4 encoding to text sets	111	cient than inspecting the \meaning	70
add T5 encoding to text sets	111	\MT@maybe@do: redone	47
declare qpl and qtm (qfonts) as aliases of ppl		\MT@pdfTeX@no: compatibility with TeXLive hack (re-	
resp. ptm	112	ported by <i>Herbert Voß</i>)	32
Inheritance: add list for OT4	115	\MT@rem@from@clist: new macro: remove an item	
add list for T5 (requested by <i>Hàn Thế Thành</i>)	116	from a comma list	40
Protrusion: fix: remove uppercase Greek letters from		\MT@scale@factor: generalised	52
T1 encoded CMR	122	\MT@toks: use instead of \toks@	34

Version 1.9a (2005/12/05)

General: new option: defersetup, by default true	98	\MT@get@highlevel: no longer check whether de-	
remove superfluous test whether \pickup@font		faults have changed	81
has changed	76	\MT@ifdefined@COT: new macros: true case only	36
Documentation: add explanation for error message		\MT@ifint: use \pdfmatch if available	37
in DVI mode	24	\MT@ifstreq: use \pdfstrcmp if available	38
add explanation for error message with non-		\MT@in@clist: fix	39
Type 1 fonts	24	\MT@info@missing@char: info instead of warning	
Font Sets: declare mdbch (mathdesign) as an alias of		(after <i>Michael Hoppe</i> reported that the ‘fl’	
Charter	112	ligature is missing in Palatino SC)	52
Protrusion: fix: remove _ from OT1 encoding	122	\MT@is@feature: new macro: check for pdfTeX fea-	
settings for T5 encoded Charter	118	ture	42
\microtypesetup: inside the preamble, accepts all		\MT@map@clist@n: following L ^A T _E X3	39
package options	102	\MT@permute@CCCC: don’t define permutations for	
\MT@check@font@cx: optimise context-sensitive setup	77	unused encodings	94
\MT@define@opt@keys: ‘(file name)/(line number)’ as		\MT@rem@from@clist: fix	40
default list name	89	\MT@setup@: defer setup until the end of the pream-	
\MT@define@set@key@: don’t expand variables imme-		ble	42
diately (requested by <i>Georg Verwey</i>)	80		

Version 1.9b (2006/01/20)

General: compatibility with listings: sanitise more catcodes (reported by <i>Holger Uhr</i>)	45	add samples of micro-typographic features	4
compatibility with the extendedchar option of the listings package	45	\MT@features: use throughout the package to adjust to beta-ness	42
Documentation: activate expansion in the distributed PDF	1	\MT@ifdimen: use \pdfmatch if available	38
		\MT@warn@code@too@large: fix calculation with present factor	52

Version 1.9c (2006/02/02)

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verweyen</i>)	20	<code>\MT@define@code@key@font</code> : fix: context was ignored	88
Protrusion: settings for URW Garamond	119	<code>\MT@define@code@key@size</code> : fix: embrace <code>\MT@tempsize</code> in <code>\csname</code> (bug introduced in 1.9b)	88

Version 1.9d (2006/05/05)

Font Sets: <code>md*</code> instead of <code>m</code> series in basic sets	111	<code>\MT@get@font@dimen</code> : warning for zero fontdimen	52
add QX encoding to text sets	111	<code>\MT@get@opt</code> : optimise: don't reset when preset option is set	53
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i>)	116	set list name before presetting	53
Protrusion: settings for QX encoding (contributed by <i>Maciej Eder</i>)	124	<code>\MT@is@active</code> : support for Unicode (inputenc/utf8)	72
settings for Euro symbols (Adobe, ITC, marvosym)	155	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>tex4ht</code> is loaded (reported by <i>Peter Dyballa</i>)	44
tweak AMS settings	147	<code>\SetProtrusion</code> : (et al.) optimise: unify keys for mandatory argument	86
<code>\DeclareCharacterInheritance</code> : fix: empty context	91	(et al.) split keys of optional and mandatory argument	86
<code>\MT@detokenize@n</code> : new macro: use <code>\detokenize</code> if available	37		
<code>\MT@get@ex@opt</code> : fix: evaluate preset	57		

Version 1.9e (2006/07/28)

General: fix: default value for <code>activate: true</code>	97	fix: forgotten comma in <code>mt-mvs.cfg</code> ; adapt to marvosym's changed encoding	155
no tracing with <code>trace</code> package	76	settings for Euler Roman font	150
Documentation: add hint about unknown encodings	23	<code>\DeclareCharacterInheritance</code> : new key 'inputenc' to set the input encoding	91
include <code>LPPL</code>	174	<code>\MT@rem@from@clist</code> : model after <code>\@remove@element</code>	40
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>)	112	<code>\MT@setup@</code> : empty <code>\MT@setup@</code> after use (compatibility with the combine class)	42
Inheritance: adapt to marvosym's changed encoding	117	<code>\SetExpansion</code> : new key: <code>inputenc</code>	86
Protrusion: complete settings for Euler Fraktur and Script fonts	154	<code>\SetProtrusion</code> : new key: <code>inputenc</code>	86

Version 1.9f (2006/09/09)

Protrusion: fix: <code>euler-vm</code> did not load euler settings	151	<code>\MT@set@inh@list</code> : fix: forgotten comma in the features list	92
<code>\MT@curr@list@name</code> : fix: <code>\MessageBreak</code> must not be expanded	73	<code>\MT@set@inputenc</code> : fix: input encoding must be set after the inheritance list has been parsed	54
<code>\MT@gdef@n</code> : new macros: global variants	36	<code>\MT@set@named@keys</code> : new macro: set name first, simplify parsing of optional argument	88
<code>\MT@glet</code> : new macro	36	<code>\SetProtrusion</code> : (et al.) set catcodes before parsing optional argument	86
<code>\MT@reset@context</code> : only reset context if it has actually been changed	78		

Version 2.0 (2007/01/14)

General: compatibility with listings: set catcode of backslash to zero (reported by <i>Steven Bath</i>)	45	(beta:1)	99
compatibility with <code>soul</code> : register <code>\textls</code> and <code>\lsstyle</code> (beta:3)	45	new package <code>letterspace</code> : a stripped-down version	1
maybe disable <code>\MT@noligatures</code> after the preamble	109	option 'babel': disable Turkish babel's active characters (beta:8)	110
new option: <code>babel</code> , by default false (language-dependent setup suggested by <i>Ulrich Dirr</i>) (beta:1)	98	option 'babel': fix: switch off French babel's active characters with <code>\shorthandoff</code> instead of <code>\NoAutoSpaceBeforeFDP</code> (reported by <i>Daniel Flipo</i>) (beta:8)	110
new option: <code>letterspace</code> , by default 100		option 'unit', <code>\SetProtrusion</code> : deprecate value	

‘relative’ completely	90	fix: letterspacing commands may be nested (beta:8)	62
option ‘letterspacing’ renamed to ‘letterspace’	99	new command: letterspacing (beta:1)	62
Documentation: add hint about how to increase font_max and font_mem_size	24	totally redone, using the new \letterspacefont	62
add hint about warning when tracking and expansion is applied to a font	24	\MT@declare@sets: fix: empty size list when redefining set	80
add remark about ‘draft’ option disabling microtype (noted by <i>Michalis Miatidis</i>)	8	\MT@is@symbol: made even more robust	72
qualify hint about web documents with regard to older pdfTeX versions	22	\MT@pdfTeX@no: case 6: pdfTeX 1.40 (beta:1)	32
qualify hints about expansion error messages with regard to older pdfTeX versions	24	\MT@set@inputenc: sanitise catcodes before loading input encoding (problem with listings)	54
Font Sets: add footnotesize and scriptsize sets	111	\MT@split@name: adjust to possible letterspacing	47
add smallcaps set	111	\SetExtraKerning: new command: additional kerning (beta:1)	87
Kerning: remove letterspacing settings	159	\SetExtraSpacing: new command: adjustment of interword spacing (beta:1)	87
\DeclareMicrotypeBabelHook: new command: interaction with babel (beta:1)	86	\SetTracking: new command: tracking	87
\lsstyle: fix: font switches don’t pose a problem anymore (beta:8)	62	\textls: new command: letterspacing (beta:1)	64
		starred version: remove spaces around text	64
		\tracingmicrotypeinpdf: new debug method: mark all fonts with PDF annotations	31

Version 2.1 (2007/01/21)

General: compatibility with CJK: also check for its definition	75	\lslig: new command: protect ligatures in letterspaced text	63
compatibility with pinyin: disable microtype in \py@macron (reported by <i>Sven Naumann</i>)	45	\MT@get@ls@basefont: redone: use \pdfmatch to make it bullet-proof	63
fix: letterspace package forgot to load keyval	34	\textls: fix: use \hmode@bgroup	64

B Index

Links refer to the page where the corresponding entry is described (bold face) resp. occurs. Plain numbers refer to the code line where the corresponding entry is defined (underlined) resp. used.

Options:		\DeclareMicrotypeSet	9
DVIoutput	8	\DeclareMicrotypeSet*	9
activate	6	\DeclareMicrotypeSetDefault	11
auto	7	\DisableLigatures	22
babel	9	\LoadMicrotypeFile	19
config	9	\lslig	21
draft	8	\lsstyle	21
expansion	6	\microtypecontext	20
factor	7	\microtypesetup	9
final	8	\SetExpansion	14
kerning	6	\SetExtraKerning	17
letterspace	8	\SetExtraSpacing	16
protrusion	6	\SetProtrusion	12
selected	8	\SetTracking	15
shrink	7	\textls	21
spacing	6	\textls*	21
step	7	\UseMicrotypeSet	11
stretch	7		
tracking	6	\@enc@update	679
unit	7		
verbose	9	A	
		a0poster (package)	82
User Commands:		activate (option)	6, 97, 166
\DeclareCharacterInheritance	18	\add@accent	1955, 1956
\DeclareMicrotypeAlias	19	ae (package)	19, 112
\DeclareMicrotypeBabelHook	21	amssymb (package)	147

- article (package) 20
 auto (option) 7, 98
- B**
- babel (option) 9, 21, 26, 98, 166
 babel (package) 3,
 5, 21, 44, 66, 86, 109, 110, 113, 163, 165–167
- C**
- \cf@encoding 679
 chemsym (package) 163
 CJK (package) 45, 167
 cm-super (package) 7
 color (package) 8, 105
 combine (package) 42, 166
 config (option) 9, 18, 27, 100, 164
 contour (package) 105
 crop (package) 105
 csquotes (package) 25, 27, 44, 122, 165
 \curr@fontshape 1457, 1905, 1914, 1916
- D**
- \DeclareCharacterInheritance .. 12, 18, 2630, 3248
 \DeclareMicrotypeAlias 13, 19, 37, 2341, 3249
 \DeclareMicrotypeBabelHook 15, 21, 31, 2374
 \DeclareMicrotypeSet .. 3, 9, 26, 27, 38, 2064, 3241
 \DeclareMicrotypeSet* 9, 2064
 \DeclareMicrotypeSetDefault 5, 11, 29, 2313
 defersetup (option) 44, 98, 102, 165
 \define@newfont 1913, 1915, 1918
 \dimexpr 1422, 1463
 \DisableLigatures 11, 22, 30, 2298
 \do@subst@correction 1904
 docstrip (package) 29
 draft (option) 8, 98, 99, 167
 dsfont (package) 50
 DVInoutput (option) 8, 27, 98, 163
- E**
- eco (package) 19, 112
 \efcode 1124, 1158, 1159, 1203, 1206
 \egroup 1514
 eucal (package) 152
 eufrak (package) 154
 euler (package) 73, 150, 166
 eulervm (package) 19, 112, 151, 166
 euroitc (package) 155
 europs (package) 155
 eurosans (package) 155
 expansion (option) 6, 11, 97, 105
- F**
- \f@encoding 679
 \f@size 1457, 1905, 1914, 1916
 factor (option) 7, 13, 27, 99, 100, 163
 fancyvrb (package) 23
 final (option) 8, 28, 98, 99, 163
 fix-cm (package) 15
 \font 672, 1486
 \font@name ... 82, 1462, 1475, 1479, 1483, 1486,
 1913, 1915, 1918, 1942, 1944, 1965, 1985
 \fontcharwd 895
- \fontdimen 837,
 844, 915, 920, 1025, 1028, 1424, 1463, 1464
 fontinst (package) 158
 fontinstallationguide (package) 137
 french (package) 164
 frenchpro (package) 72, 163
- G**
- german (package) 33, 162
 graphics (package) 8, 105
- H**
- hfoldsty (package) 19, 112
 \hmode@bgroup 1503
 hyperref (package) 8, 44, 75, 105, 164, 165
- I**
- IEEEtran (package) 75
 \iffontchar 925
 \ifMT@auto 180, 3310, 3355
 \ifMT@babel 180, 3443, 3490
 \ifMT@do 706, 789, 1540
 \ifMT@document 222, 2750
 \ifMT@draft 180, 3230
 \ifMT@expansion 180, 3274, 3352
 \ifMT@if@ 179, 1979, 2860, 3510
 \ifMT@inannot 71
 \ifMT@inlist@
 .. 423, 468, 536, 682, 734, 756, 773, 782,
 802, 1574, 1585, 1676, 1973, 2037, 2049, 2359
 \ifMT@kerning 180, 3394, 3430, 3503
 \ifMT@noligatures 180, 3459
 \ifMT@nonselected 1105, 1689, 1697
 \ifMT@noreset 1753, 1760
 \ifMT@opt@auto 2867, 3313
 \ifMT@opt@DVI 2867, 3260
 \ifMT@opt@expansion 2867, 3255
 \ifMT@protrusion 180, 563, 585, 3262
 \ifMT@selected 180, 3338, 3358
 \ifMT@spacing 180, 3385, 3422, 3442
 \ifMT@tracking 180, 1428, 3377, 3414
 ifpdf (package) 8, 105
 \iftracingmicrotypeinpdfall 76
 inputenc (package) 14, 23, 26, 45, 54, 70–73, 164, 166
 \inputencoding 1037, 1041
- J**
- jurabib (package) 75
- K**
- kerning (option) 6, 11, 26, 97
 keyval (package) 34, 50, 80, 113, 167
 \knacode 1365, 1366, 1378, 1382, 1389
 \knbccode 1358, 1359, 1377, 1381, 1388
 \knbscode 1250, 1251, 1276, 1281, 1289
- L**
- ledmac (package) 25, 27, 43, 75, 164
 ledpar (package) 25
 \leftmarginkern 571
 letterspace (option)
 8, 16, 21, 22, 26, 37, 62, 99, 166, 167
 letterspace (package) ... 1, 22, 26, 29, 43, 166, 167

- \letterspacefont 1462
 lineno (package) 43
 listings (package) 23, 45, 54, 66, 165–167
 lmodern (package) 7, 124
 \LoadMicrotypeFile 14, 19, 40, 2355, 3250
 \lcode 830, 868, 869, 1051, 1055
 \lslig 21, 23, 1476
 \lssstyle .. 20, 21, 626, 651, 1413, 1494, 1498, 1512
- M**
- marvosym (package) 18, 19, 26, 29, 117, 155, 166
 mathdesign (package) 19, 112, 165
 memoir (package) 20, 75
 \Microtype@Hook 101, 3086
 \microtypecontext .. 17, 20, 2007, 3252, 3473, 3476
 \microtypesetup 9, 16, 3090, 3251
 minimal (package) 46
 \MT@font 256, 685, 757, 760,
 839, 916, 927, 934, 1467, 1631, 1702, 1945
 \MT@abbr@ex 512
 \MT@abbr@ex@c 512
 \MT@abbr@ex@inh 512
 \MT@abbr@kn 512
 \MT@abbr@kn@c 512
 \MT@abbr@kn@inh 512
 \MT@abbr@nl 512
 \MT@abbr@pr 512
 \MT@abbr@pr@c 512
 \MT@abbr@pr@inh 512
 \MT@abbr@sp 512
 \MT@abbr@sp@c 512
 \MT@abbr@sp@inh 512
 \MT@abbr@tr 512
 \MT@abbr@tr@c 512
 \MT@active@features
 1967, 1970, 1982, 1989, 2025, 2036, 2300,
 3263, 3353, 3378, 3386, 3395, 3465, 3466
 \MT@addto@annot 71
 \MT@addto@setup
 546, 603, 1902, 2007, 2008, 2975, 3091,
 3229, 3375, 3413, 3458, 3464, 3489, 3520
 \MT@auto 1114, 3309, 3322, 3331, 3335
 \MT@auto@ 1102, 1114, 1118, 1176
 \MT@autofalse 182, 3320, 3330
 \MT@autotrue 182, 3050
 \MT@babelfalse 189, 3239
 \MT@babeltrue 189
 \MT@begin@catcodes 1575, 1576, 1614, 2363, 2381,
 2395, 2417, 2431, 2445, 2635, 3063, 3064
 \MT@cat 1034, 1895, 1897
 \MT@catcodes 153, 173
 \MT@char 858, 868,
 869, 874, 875, 878, 880, 895, 900, 925,
 1051, 1052, 1054–1056, 1146, 1158, 1159,
 1161, 1163, 1203, 1205, 1206, 1239, 1250,
 1251, 1257, 1258, 1264, 1265, 1268, 1270,
 1276–1278, 1280–1283, 1347, 1358, 1359,
 1365, 1366, 1369, 1371, 1377, 1378, 1380–
 1382, 1733, 1749, 1750, 1755, 1825, 1828,
 1830, 1856, 1861, 2695, 2696, 2701, 2704
 \MT@char@ 1733,
 1739, 1744, 1749, 1764, 1766, 1772, 1773,
 1775, 1788, 1789, 1792, 1793, 1796, 1797,
 1801, 1803, 1829, 1842, 1845, 1848, 1869
 \MT@charstring 1746, 1828, 1833
 \MT@check@active@set
 3078, 3269, 3359, 3380, 3389, 3399
 \MT@check@font 681, 1964, 2027
 \MT@check@font@cx 1968, 2027
 \MT@check@range 2831, 2833
 \MT@check@range@ 2833, 2834
 \MT@check@rlist 2780, 2824
 \MT@check@rlist@ 2824, 2825
 \MT@checklist@ 713, 726, 1537
 \MT@checklist@family 766
 \MT@checklist@font 745
 \MT@checklist@size 795
 \MT@checksetup 3101, 3112, 3119, 3140, 3162
 \MT@clist@break 395, 740, 758, 790, 808
 \MT@clist@function 395
 \MT@cnt@encoding 2716, 2724, 2725
 \MT@cnt@family 2722, 2731, 2732
 \MT@cnt@series 2729, 2738, 2739
 \MT@cnt@shape 2736, 2744, 2745
 \MT@config@file 3053, 3061, 3062, 3066, 3067, 3070
 \MT@context 1652, 1664, 1683, 1710
 \MT@count 346, 347, 485, 887, 895,
 896, 901, 902, 907, 908, 912, 920, 957, 1019
 \MT@curr@file 1579, 1580, 1587,
 1588, 2365, 2366, 2535, 2657, 3066, 3522
 \MT@curr@list@name 918, 960, 1044, 1172,
 1560, 1723, 1864, 1872, 1877, 1885, 1891
 \MT@curr@set@name
 2094, 2096–2099, 2101, 2102, 2115,
 2120, 2124, 2125, 2150, 2154, 2183, 2214,
 2220, 2225, 2229, 2230, 2485, 2504, 2542,
 2552, 2556, 2560, 2569, 2572, 2577, 2582,
 2585, 2588, 2604, 2620, 2626, 2658, 2660
 \MT@declare@char@inh 2640, 2645, 2655
 \MT@declare@sets 2081, 2087, 2095, 2302
 \MT@DeclareSet 2068, 2069, 2071
 \MT@DeclareSet@ 2072, 2075, 2078
 \MT@DeclareSetAndUseIt 2066, 2067, 2071
 \MT@def@bool@opt 2943, 2959, 2961, 2970, 2979
 \MT@def@n 245, 2060, 2061
 \MT@def@num@opt 3005, 3019, 3020
 \MT@def@simple@bool@opt 2959, 2960
 \MT@default@ex@set 2329
 \MT@default@kn@set 2329
 \MT@default@pr@set 2329
 \MT@default@sp@set 2329
 \MT@default@tr@set 2329
 \MT@define@code@key . 2467, 2526–2529, 2680–2682
 \MT@define@code@key@font 2490, 2531, 2684
 \MT@define@code@key@size 2478, 2530, 2683
 \MT@define@context 2032, 2058
 \MT@define@ex@opt@key 2600, 2613–2615
 \MT@define@key@unit 2580, 2598, 2599
 \MT@define@opt@key 2544–2547, 2550
 \MT@define@opt@keys 2532, 2554
 \MT@define@option 2870, 2906, 2907
 \MT@define@option@ 2912, 2940–2942
 \MT@define@optionX 3090, 3150, 3151
 \MT@define@optionX@ 3152, 3183, 3185, 3186
 \MT@define@set@key@ 2106–2109, 2113

- \MT@define@set@key@font 2111, 2218
 \MT@define@set@key@size 2110, 2143
 \MT@define@set@keys 2105, 2266, 2267
 \MT@detokenize@c 296, 1741, 1826
 \MT@detokenize@n 296, 1858, 1859
 \MT@dimen@six 836, 888, 912, 957
 \MT@dinfo 60
 \MT@dinfo@nl 60
 \MT@do@font 484, 829, 1124, 1288, 1387
 \MT@documentfalse 222
 \MT@documenttrue 222, 3468
 \MT@dofalse 706, 717, 739, 761, 778, 807
 \MT@dotrue 706, 709, 736, 775, 784, 804, 1533
 \MT@draftfalse 185, 2981
 \MT@drafttrue 185, 2983
 \MT@edef@n 247, 1688, 1714, 2053,
 2473, 2535, 2537, 2656, 2671, 2811, 3012
 \MT@encoding 696, 1647,
 1659, 1741, 1826, 1857, 1876, 1883, 1891
 \MT@encoding@check 2669, 2673
 \MT@end@catcodes 1577, 1619, 2371,
 2392, 2414, 2428, 2442, 2456, 2650, 3065
 \MT@error 42, 539, 552, 1044, 3485
 \MT@ex@cc@name 1098, 1100,
 1178, 1179, 1189, 1195, 1196, 1210, 2398
 \MT@ex@context 2019, 2059
 \MT@ex@do 1101, 1136
 \MT@ex@doc@contexts 2059
 \MT@ex@factor 190, 1115, 1182
 \MT@ex@factor@ 1115, 1126, 1129, 1148, 1149, 1176
 \MT@ex@inh@name 1160, 1161, 1163
 \MT@ex@level 190, 3354, 3356
 \MT@ex@max 202, 1151, 1152
 \MT@ex@min 202, 1154, 1155
 \MT@ex@setname 2284
 \MT@ex@split 1138, 1142
 \MT@exp@one@n ... 257, 402, 535, 676, 678, 731,
 750, 1584, 1964, 1971, 2003, 2036, 2047,
 2059, 2087, 2277, 2322, 2358, 2645, 3473
 \MT@exp@two@c 259, 287,
 291, 302, 437, 675, 1475, 1740, 1745, 1827
 \MT@exp@two@n 261, 771, 780, 2119, 2224
 \MT@expansion 687, 1091, 3369
 \MT@expansionfalse 181, 3235, 3256, 3348
 \MT@expansiontrue 181, 3049
 \MT@extra@context
2059, 2384, 2398, 2421, 2434, 2448, 2495,
 2499, 2500, 2503, 2548, 2631, 2777, 2783,
 2787, 2791, 2794, 2799, 2800, 2803, 2830
 \MT@extra@inputenc 2630, 2654, 2659, 2660
 \MT@factor@default 214, 3024, 3030, 3266
 \MT@family 676, 696, 772, 1648, 2350, 2351
 \MT@familyalias 677,
 678, 702, 779, 781, 783, 785, 1658, 1660, 2352
 \MT@feat 706, 841, 939, 941, 943,
 944, 946, 947, 953, 955, 958, 966–970,
 972, 974–978, 981, 982, 987, 991, 994,
 997, 1000–1002, 1011, 1012, 1017, 1036,
 1040, 1042, 1550, 1552, 1555, 1556, 1560,
 1562, 1563, 1565, 1684, 1686, 1688, 1692,
 1696, 1700, 1701, 1704, 1712, 1714, 1716,
 1719, 1726, 1730, 1865, 1866, 1895, 1897
 \MT@features 532, 2059,
 2081, 2266, 2271, 2316, 2554, 2640, 3074
 \MT@features@long 532, 535, 540, 2058, 2652
 \MT@file@list 1571,
 1573, 1581, 1584, 1589, 1592, 2358, 2362
 \MT@find@file 676, 678, 1571
 \MT@fix@font@set 2138, 3261
 \MT@font 256, 675, 694, 754, 830, 831, 837,
 844, 868, 869, 874, 875, 895, 900, 915,
 920, 925, 1025, 1028, 1051, 1052, 1055,
 1056, 1102, 1118, 1124, 1158, 1159, 1203,
 1206, 1250, 1251, 1257, 1258, 1264, 1265,
 1276–1278, 1281–1283, 1289–1291, 1358,
 1359, 1365, 1366, 1377, 1378, 1381, 1382,
 1388, 1389, 1456, 1469, 1541, 1900, 1941,
 1942, 1944, 1964, 1966, 1972, 1993, 2004
 \MT@font@list 1899, 1964–1966, 2026
 \MT@font@sets 2123, 2138, 2228, 3261
 \MT@gdef@n 245, 2377, 2391, 2413, 2427,
 2441, 2455, 2577, 2604, 2620, 2662, 3076
 \MT@get@axis 2237–2240, 2253, 2512–2515
 \MT@get@basefamily 1583, 1620
 \MT@get@char@unit 859, 989, 1015, 1240, 1348
 \MT@get@charwd 893, 912, 992, 1015
 \MT@get@config 3053
 \MT@get@ex@opt 1094, 1109, 1176
 \MT@get@ex@opt@ 1184–1186, 1188, 1194
 \MT@get@font 2223, 2233
 \MT@get@font@and@size 2494, 2508
 \MT@get@font@dimen 914, 995
 \MT@get@font@dimen@six . 816, 836, 1218, 1326, 1443
 \MT@get@highlevel ... 2118, 2128, 2255, 2472, 2670
 \MT@get@inh@list 819, 1096, 1221, 1329, 1710
 \MT@get@listname 1630, 1686, 1712
 \MT@get@listname@ 1630
 \MT@get@ls@basefont 1460, 1478, 1481
 \MT@get@opt 817, 964, 1219, 1327
 \MT@get@range 2147, 2157, 2482
 \MT@get@size .. 2164, 2169, 2178, 2197, 2247, 2523
 \MT@get@slot 857, 1145, 1238, 1346, 1733, 2694, 2700
 \MT@get@space@unit 989, 1081,
 1248, 1255, 1262, 1309, 1313, 1356, 1363
 \MT@get@unit 997, 1005, 1445
 \MT@get@unit@ 1005
 \MT@glet 242, 547,
 694, 1498, 1937, 2027–2030, 2310, 2352, 2719
 \MT@gobble@settings 836
 \MT@if@false 179, 1976, 2828, 3504
 \MT@if@list@exists
 ... 815, 1093, 1108, 1217, 1325, 1437, 1683
 \MT@if@true
179, 1969, 2837, 2841, 2849, 2854, 3505–3509
 \MT@ifdefined@c@T ... 265, 596, 610, 877, 1160,
 1267, 1368, 2350, 2659, 2766, 3086, 3089
 \MT@ifdefined@c@TF
265, 564, 1037, 1422, 1434, 1687, 1713, 2776
 \MT@ifdefined@n@T ... 265, 728, 747, 768, 797,
 878, 1000, 1035, 1161, 1189, 1268, 1369,
 1440, 1666, 1894, 2097, 2345, 2401, 2538,
 2725, 2732, 2739, 2745, 2777, 2791, 3212
 \MT@ifdefined@n@TF 265, 702, 711, 729,
 748, 769, 798, 966, 974, 1178, 1195, 1535,

- 1555, 1654, 1741, 2286, 2290, 2331, 2752,
2810, 2819, 2889, 2925, 3075, 3079, 3471
\MT@ifdim 355,
457, 458, 462, 463, 2180, 2189, 2835–
2837, 2839, 2840, 2847–2849, 2852, 2853
\MT@ifdimen 334, 2209, 2559, 2571, 2587, 3038
\MT@ifempty 304, 866,
872, 1247, 1254, 1261, 1355, 1362, 1508,
2038, 2080, 2085, 2130, 2136, 2158, 2159,
2172, 2173, 2234, 2241, 2256, 2270, 2275,
2315, 2320, 2509, 2517, 2534, 2548, 2551,
2602, 2639, 2644, 2674, 2875, 2917, 2945,
2988, 3008, 3023, 3034, 3055, 3098, 3159
\MT@ifint 316, 1795, 2603, 3011, 3026
\MT@ifstreq 362, 1551,
1625–1627, 1685, 1700, 1761, 2002, 2558,
2584, 2586, 2878, 2880, 2883, 2886, 2920,
2922, 2948, 2949, 2991, 2992, 2997, 3037,
3100, 3107, 3111, 3118, 3161, 3167, 3445
\MT@in@clist 423, 535, 731, 751,
771, 780, 1573, 1584, 1964, 1971, 2036, 2358
\MT@in@rlist 451, 800, 1675
\MT@in@rlist@ 451
\MT@in@rlist@ 451
\MT@in@tlist 439, 2047
\MT@in@tlist@ 439
\MT@inannotfalse 71
\MT@inannottrue 71
\MT@increment 485, 2724, 2731, 2738, 2744
\MT@info 42, 63, 68, 2288, 3103,
3109, 3115, 3122, 3164, 3169, 3199, 3203
\MT@info@missing@char 896, 902, 923
\MT@info@nl 42, 64, 65, 69, 565, 925,
932, 2987, 3062, 3080, 3083, 3259, 3265,
3272, 3355, 3370, 3379, 3383, 3388, 3392,
3398, 3402, 3418, 3426, 3434, 3481, 3492
\MT@inh@do 1725, 2685
\MT@inh@feat 2630, 2639, 2642, 2653
\MT@inh@split 2687, 2691
\MT@inlist@false 423, 427, 440, 452, 1979
\MT@inlist@true 423, 429, 447, 459, 464, 1979
\MT@is@active 1740, 1808
\MT@is@char 1745, 1827, 1833
\MT@is@composite 1743, 1854
\MT@is@feature 534, 2086, 2276, 2321
\MT@is@letter 1738, 1762, 1830, 1861
\MT@is@number 1778, 1783
\MT@is@symbol 1742, 1824
\MT@iterate 473
\MT@kerning 689, 1322, 1410, 3401, 3437
\MT@kerningfalse 187, 3238, 3436
\MT@kerningtrue 187
\MT@kn@cc@name 1331, 1333, 1396, 2448
\MT@kn@context 2020, 2059
\MT@kn@do 1334, 1337
\MT@kn@doc@contexts 2059
\MT@kn@factor 190
\MT@kn@factor@ 966
\MT@kn@inh@name 1368, 1369, 1371
\MT@kn@max 202
\MT@kn@min 202
\MT@kn@setname 2284
\MT@kn@split 1339, 1343
\MT@kn@split@val 1349, 1353
\MT@kn@unit 190
\MT@kn@unit@ 974, 1399
\MT@led@kern 561
\MT@led@unhbox@line 561
\MT@ledmac@setup 561, 606
\MT@let@cn 254, 255, 703, 823, 1100, 1179,
1225, 1333, 1439, 1441, 1444, 1550, 1562,
1680, 1684, 1719, 2201, 2542, 2658, 3009
\MT@let@nc
. 251, 560, 1002, 1420, 1696, 1726, 1730,
1974, 2026, 2099, 2101, 2115, 2220, 2406,
2556, 2560, 2569, 2572, 2582, 2585, 2588,
2626, 2812, 2817, 2884, 2887, 3114, 3121
\MT@let@nn 255,
967, 972, 975, 987, 1196, 1199, 2046, 3082
\MT@letterspace . 218, 1423, 1435, 3405, 3406, 3408
\MT@letterspace@ 1422, 1423, 1434, 1435, 1439,
1448, 1454, 1458, 1462, 1463, 1467, 1507
\MT@letterspace@default 218, 3406
\MT@listname 1630,
1677, 1680, 1687, 1688, 1690, 1692, 1713,
1714, 1717, 1719, 1723, 1726, 2703, 2710
\MT@load@list 821, 1098, 1223, 1331, 1548
\MT@loop 473, 480, 2816
\MT@lower 2151, 2157, 2485
\MT@ls@adjust 1425, 1504, 1505, 1513, 1528
\MT@ls@adjust@ 1504, 1528
\MT@ls@too@large 1511, 1516, 3408
\MT@lsfont 1424, 1455, 1462–1464, 1469, 1475
\MT@lskern 1422, 1479, 1527, 1529
\MT@lslig 1420, 1476
\MT@make@string 511, 2141, 2344, 2357
\MT@map@clist@ 395
\MT@map@clist@c . 395, 1970, 1982, 1989, 2025,
2058, 2081, 2083, 2266, 2271, 2273, 2316,
2318, 2554, 2640, 2642, 2652, 3074, 3210
\MT@map@clist@n 395, 710,
1534, 2059, 2116, 2145, 2221, 2375, 2470,
2480, 2492, 2697, 2873, 2915, 3095, 3156
\MT@map@tlist@ 413
\MT@map@tlist@c 413, 442, 453, 879, 1162,
1269, 1370, 1634, 1995, 2778, 2829, 3261
\MT@map@tlist@n . 413, 1610, 1835, 2718, 2960, 3019
\MT@maybe@do 706, 813, 1091, 1215, 1323, 1428
\MT@MT 41, 43, 46, 56, 58, 59, 63, 64, 66, 67,
140, 553, 555, 3056, 3199, 3203, 3260, 3449
\MT@next@listname 1637, 1646
\MT@next@listname@ 1655, 1667, 1673
\MT@nl@setname 2303
\MT@noligatures 690, 1531, 3460
\MT@noligaturesfalse 184
\MT@noligaturestrue 184, 2301
\MT@nonselectedfalse 1105, 1119
\MT@nonselectedtrue 1105, 1107
\MT@noresetfalse 1760, 1761, 1768, 1777, 1798
\MT@noresettrue 1736, 1760, 1786
\MT@normal@catcodes 638, 1038, 1599, 1616
\MT@old@cmd 32, 37–40
\MT@opt@autofalse 2868
\MT@opt@autotruet 2868
\MT@opt@DVIfalse 2869
\MT@opt@DVIttrue 2869, 2963, 2966

- `\MT@opt@expansionfalse` 2867
`\MT@opt@expansiontrue` 2867
`\MT@options` 2458
`\MT@orig@add@accent` 1955
`\MT@orig@foreign@language` 3498, 3500
`\MT@orig@pickupfont` 625, 659, 664, 1908, 1957
`\MT@orig@select@language` 3493, 3495
`\MT@pdf@annot` 71
`\MT@pdfcprot@error` 551, 605
`\MT@pdf@tex@no` 93, 130, 131, 135, 236, 3334
`\MT@permute` 2390, 2412, 2426, 2440, 2454, 2665, 2715
`\MT@permute@` 2715
`\MT@permute@@` 2715
`\MT@permute@@@` 2715
`\MT@permute@@@` 2715
`\MT@permute@@@` 2743, 2748
`\MT@permute@@@` 2756, 2758
`\MT@permute@define` 2749, 2759–2761, 2808
`\MT@permute@reset` 2718, 2814
`\MT@permutelist`
 2384, 2398, 2424, 2434, 2448, 2495, 2496,
 2503, 2663, 2772, 2773, 2777, 2783, 2787,
 2791–2794, 2803, 2804, 2830, 2861, 2862
`\MT@pickupfont` 661, 666, 1954, 1959
`\MT@pr@c@name` 821, 823, 1061, 2384
`\MT@pr@context` 2019, 2059
`\MT@pr@do` 824, 848
`\MT@pr@doc@contexts` 2059
`\MT@pr@factor` 190, 3027, 3266, 3267
`\MT@pr@factor@` 966
`\MT@pr@inh@name` 877, 878, 880
`\MT@pr@level` 190, 3264, 3265
`\MT@pr@max` 202
`\MT@pr@min` 202
`\MT@pr@setname` 2284
`\MT@pr@split` 850, 854
`\MT@pr@split@val` 860, 864
`\MT@pr@unit` 190, 3039, 3268
`\MT@pr@unit@` 974, 1064
`\MT@preset@aux` 1066, 1068,
 1070, 1071, 1074, 1401, 1403, 1405, 1406
`\MT@preset@aux@factor` 1066, 1074, 1303–1305, 1401
`\MT@preset@aux@space` 1068, 1074, 1307, 1403
`\MT@preset@ex` 1190, 1209
`\MT@preset@kn` 1394
`\MT@preset@kn@` 1394
`\MT@preset@pr` 1059
`\MT@preset@pr@` 1059
`\MT@preset@sp` 1296
`\MT@preset@sp@` 1296
`\MT@ProcessOptionsWithKV` 3206, 3227
`\MT@protrusion` 686, 813, 3271
`\MT@protrusionfalse` 180, 3234
`\MT@protrusiontrue` 180, 3046
`\MT@py@macron` 656, 660, 665
`\MT@rba@expansion` 527
`\MT@rba@kerning` 527
`\MT@rba@protrusion` 527
`\MT@rba@spacing` 527
`\MT@rba@tracking` 527
`\MT@register@font` 692, 1966, 2028
`\MT@register@font@cx` 1988, 2028
`\MT@register@subst@font` 1946, 1965, 2029
`\MT@register@subst@font@cx` 1981, 2029
`\MT@rem@from@clist` 434, 2003, 2462
`\MT@rem@from@lists` 1997, 2001
`\MT@repeat` 473, 482, 2822
`\MT@requires@etex` 223, 265, 296, 486, 495, 893, 923
`\MT@requires@pdftex` 235, 316, 334, 362, 561, 885,
 1127, 1214, 1322, 1413, 1531, 2298, 2619,
 2625, 3048, 3152, 3311, 3373, 3440, 3457
`\MT@reset@context` 2011, 2014, 2017, 2045
`\MT@reset@context@` 2017, 2045
`\MT@reset@ef@codes` 1095, 1117, 1127, 1191
`\MT@reset@ef@codes@` 1122, 1130, 1134
`\MT@reset@kn@codes` 1328, 1335, 1385
`\MT@reset@kn@codes@` 1385
`\MT@reset@pr@codes` 818, 825, 834
`\MT@reset@pr@codes@` 834
`\MT@reset@sp@codes` 1220, 1227, 1286
`\MT@reset@sp@codes@` 1286
`\MT@restore@catcodes` 166, 174
`\MT@restore@p@h` 594, 599, 600, 623, 632
`\MT@sav@set@font` 3197, 3204
`\MT@scale` 495, 888, 940, 954, 957, 1149
`\MT@scale@factor` 890, 909, 938, 1076, 1211
`\MT@scale@to@em` 867, 873, 885, 906, 1082, 1249,
 1256, 1263, 1310, 1314, 1357, 1364, 1447
`\MT@selectedfalse` 183
`\MT@selectedtrue` 183
`\MT@series` 696, 1649, 1661
`\MT@set@all@ex` 1122, 1212
`\MT@set@all@kn` 1385, 1407
`\MT@set@all@pr` 827, 834, 1072
`\MT@set@all@sp` 1286, 1317
`\MT@set@babel@context` 3470, 3496, 3501, 3513
`\MT@set@default@set` 2316, 2322, 2329
`\MT@set@ex@codes` 1121, 3339, 3341
`\MT@set@ex@codes@n` 1105, 1121, 3341
`\MT@set@ex@codes@s` 1092, 3339
`\MT@set@ex@heirs` 1164, 1202
`\MT@set@inh@list` 2636, 2638
`\MT@set@inputenc` 820, 1033, 1097, 1222, 1330, 1724
`\MT@set@kn@codes` 1324
`\MT@set@kn@heirs` 1372, 1376
`\MT@set@listname`
 822, 965, 1099, 1177, 1224, 1332, 1438, 1864
`\MT@set@lsfont` 1472, 1475
`\MT@set@named@keys`
 2386, 2400, 2422, 2436, 2450, 2458
`\MT@set@pr@codes` 814
`\MT@set@pr@heirs` 881, 1050
`\MT@set@sp@codes` 1216
`\MT@set@sp@heirs` 1271, 1275
`\MT@set@tr@codes` 1429, 1433
`\MT@SetExpansion` 2396, 2398
`\MT@SetExtraKerning` 2446, 2448
`\MT@SetExtraSpacing` 2432, 2434
`\MT@SetProtrusion` 2382, 2384
`\MT@SetTracking` 2418, 2420
`\MT@setup@` 544, 546, 547, 2973, 2974
`\MT@setup@contexts` 2010, 2024
`\MT@setup@font` .. 673, 1949, 3197, 3200, 3204, 3240
`\MT@setup@font@hook`
 595, 607, 609, 614, 623, 632, 643, 674
`\MT@shape` 696, 1650, 1662

- \MT@shorthandoff 3479, 3510, 3511
 \MT@showpdfannot 77
 \MT@shrink 190, 1112, 3278, 3280,
 3282, 3286, 3287, 3290, 3294, 3344, 3357
 \MT@shrink@ 1102, 1112, 1118, 1176
 \MT@shrink@default 215, 3282
 \MT@size 458, 462, 463, 696, 803, 806
 \MT@size@name 451, 1677
 \MT@sp@c@name 1223, 1225, 1298, 2434
 \MT@sp@context 2020, 2059, 3445
 \MT@sp@do 1226, 1229
 \MT@sp@doc@contexts 2059
 \MT@sp@factor 190
 \MT@sp@factor@ 966
 \MT@sp@inh@name 1267, 1268, 1270
 \MT@sp@max 202
 \MT@sp@min 202
 \MT@sp@setname 2284
 \MT@sp@split 1231, 1235
 \MT@sp@split@val 1241, 1245
 \MT@sp@unit 190
 \MT@sp@unit@ 974, 1301
 \MT@spacing 688, 1214, 1320, 3391, 3429
 \MT@spacingfalse 186, 3237, 3428
 \MT@spacingtrue 186
 \MT@split@name 675, 696
 \MT@split@val 1245
 \MT@step 190, 1113, 3285, 3301, 3308, 3358
 \MT@step@ 1102, 1113, 1118, 1176
 \MT@step@default 215
 \MT@stretch 190, 1111, 3275, 3276, 3279,
 3280, 3286, 3288, 3293, 3296, 3343, 3357
 \MT@stretch@ 1102, 1111, 1118, 1176
 \MT@stretch@default 215, 3276
 \MT@temp 2236,
 2264, 2511, 3209, 3213, 3218, 3220, 3223
 \MT@tempencoding 2751, 2752, 2762, 2769
 \MT@tempfamily 2763
 \MT@tempseries 2764
 \MT@tempshape 2765
 \MT@tempsize 2719, 2766, 2776, 2779, 2784
 \MT@test@ast 2129, 2134
 \MT@testrest 1760, 1843, 1846
 \MT@textls 18, 1504, 1505, 1507
 \MT@tlist@break 413, 448, 470, 1639, 2864
 \MT@toks 178, 925, 933, 960, 1172, 1737,
 1766, 1773, 1818, 1848, 1869, 1881, 1889
 \MT@tr@c@name 1439–1441, 1444, 2423, 2427
 \MT@tr@context 2020, 2059
 \MT@tr@doc@contexts 2059
 \MT@tr@factor@ 1446
 \MT@tr@max 202, 1517–1519
 \MT@tr@min 202, 1521–1523
 \MT@tr@setname 1417, 2284
 \MT@tr@unit@ 1441, 1442, 1445
 \MT@tracking 680, 1419, 1427, 1493, 1933, 3382, 3421
 \MT@tracking@ 1419, 1427
 \MT@trackingfalse 188, 3184, 3236, 3420
 \MT@trackingtrue 188, 1416, 3184
 \MT@try@order 1634, 1642
 \MT@undefined@char 1813, 1815, 1823
 \MT@upper 2151, 2157, 2485
 \MT@use@set 2271, 2277, 2284
 \MT@val 2038–2040, 2042, 2048,
 2050, 2053, 2117, 2121, 2129, 2131, 2146–
 2148, 2160, 2163, 2165, 2168, 2170, 2171,
 2177, 2179, 2180, 2182, 2185, 2187, 2198,
 2201, 2205, 2209–2211, 2213, 2215, 2222,
 2223, 2226, 2243, 2245, 2248–2250, 2254,
 2256, 2258, 2260, 2471, 2473, 2481–2483,
 2493, 2494, 2498, 2504, 2519, 2521, 2668,
 2669, 2671, 2675, 2696, 2703, 2709, 2710,
 2874, 2875, 2878, 2880, 2883, 2886, 2889,
 2890, 2895, 2916, 2917, 2920, 2922, 2925,
 2926, 2931, 3096, 3098, 3100, 3107, 3111,
 3118, 3126, 3157, 3159, 3161, 3167, 3172
 \MT@vinfo 42, 65, 685, 723, 969,
 978, 982, 1017, 1180, 1197, 1467, 1542,
 1556, 1580, 1588, 1591, 1690, 1692, 1698,
 2018, 2042, 2360, 2366, 2987, 2996, 3472
 \MT@warn@ascii 1775, 1868
 \MT@warn@axis@empty 2242, 2257, 2262, 2518
 \MT@warn@code@too@large 944, 947, 951
 \MT@warn@err 42, 2993, 2994
 \MT@warn@ex@too@large 1152, 1155, 1170
 \MT@warn@maybe@inputenc 1882, 1890, 1893
 \MT@warn@number@too@large 1802, 1874
 \MT@warn@preset@towidth ... 1065, 1085, 1302, 1400
 \MT@warn@rest 1754, 1879
 \MT@warn@unknown 1751, 1887
 \MT@warning . 33, 42, 66, 1495, 1518, 1522, 1552,
 1565, 1701, 2098, 2181, 2213, 2263, 2293,
 2307, 2335, 2346, 2368, 2539, 2562, 2574,
 2590, 2606, 2622, 2676, 2770, 2792, 2861,
 2993, 3086, 3125, 3144, 3171, 3189–3191
 \MT@warning@nl 42,
 67, 134, 577, 586, 616, 838, 916, 958,
 1010, 1086, 1171, 1869, 1875, 1880, 1888,
 1921, 2403, 2894, 2930, 2950, 2994, 2998,
 3013, 3028, 3041, 3069, 3231, 3303, 3314,
 3325, 3345, 3415, 3423, 3431, 3446, 3515
 \MT@while@enum 478, 484, 639–641
 \MT@with@babel@and 548, 596, 608, 3505–3509, 3511
 \MT@with@package 548, 559, 597,
 600, 605, 606, 612, 631, 637, 650, 655, 1930
 \MT@xadd 75, 381, 1581, 1589, 1592,
 1983, 1991, 2050, 2149, 2362, 2484, 2702
 \MT@xaddb 388, 2502, 2782
 \MT@xdef@n
 247, 2287, 2291, 2333, 2338, 2349, 2495,
 2552, 2660, 2803, 2890, 2892, 2926, 2928
- N**
- \normalfont 672, 3362, 3366
 \nullfont 672
- P**
- \pdfadjustinterwordglue 3185, 3387
 \pdfadjustspacing 3151, 3354, 3365
 \pdfannot 80
 \pdfappendkern 3187, 3397
 pdfcpot (package) 23, 43, 51, 74, 162
 \pdfescapestring 75
 \pdffontexpand 1102, 1118
 \pdflastmatch 1487, 1488
 \pdfmatch 318, 336, 1482

C The L^AT_EX Project Public License

We, the L^AT_EX3 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 ‘`cfguide.tex`’ and ‘`modguide.tex`’ in the base L^AT_EX 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^AT_EX-Format' or in the case of files belonging to the 'L^AT_EX-format' a program implementing the 'T_EX 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

author 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^AT_EX 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 Main-

tainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.

- 5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document ‘`modguide.tex`’ in the base L^AT_EX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing L^AT_EX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L^AT_EX, 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 in-

terest 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^AT_EX-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.