



The microtype package

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

R Schlicht w.m.l@gmx.net

v1.9 — 2005/10/28

Abstract

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

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

-
- 1 A preview of the next version with support for even more micro-typographical extensions is also included in this package. Footnote 12 on page 17 contains the details.
 - 2 Currently, this package provides settings for Computer Modern Roman, Palatino, Times, Adobe Garamond and Minion, Bitstream Charter, and the AMS math fonts, as well as some generic settings for unknown fonts. Contributions are very welcome.

Contents

1	Micro-Typography with pdf\TeX	4
2	Invoking the Package	5
3	Options	5
3.1	Micro-Typographic Options	5
3.2	Options for Character Protrusion	6
3.3	Options for Font Expansion	6
3.4	Miscellaneous Options	7
3.5	Changing Settings Later	8
4	Declaring Font Sets	8
5	Micro Fine Tuning	11
5.1	Character Protrusion	11
5.2	Font Expansion	13
5.3	Character Inheritance	15
5.4	Configuration Files	15
6	Disabling Ligatures	17
7	Context-sensitive setup	17
8	Hints and Caveats	17
9	Contributions	19
10	Acknowledgments	19
11	References	20
12	Short History	20
13	Implementation	23
13.1	Requirements	25
13.2	Compatibility	28
13.3	Auxiliary macros	30
13.4	Setting up a font	36
	[Protrusion – 40] [Expansion – 45] [Interword Space (Glue) – 48]	
	[Additional Kerning – 51] [Letter spacing – 53] [Disabling	
	Ligatures – 54] [Loading the Configuration – 54] [Translating	
	Characters – 58] [Hook into \LaTeX 's font selection – 63]	
13.5	Configuration	66
	[Font Sets – 66] [Interaction with babel – 73] [Declarations – 73] [Fine	
	Tuning – 75] [Character Inheritance – 82] [Permutation – 84]	
13.6	User Command	87

13.7 Package Options	91
[Declaring the Options – 91] [Hook for Other Packages – 95] [Processing the Options – 96]	
14 Configuration Files	102
14.1 Character Inheritance	102
[OT1 – 102] [T1 – 103] [LY1 – 103] [OT4 – 104] [T5 – 105]	
14.2 Font Expansion	105
14.3 Character Protrusion	107
[Default – 107] [Italics – 113] [Small caps – 120] [Italic Small Caps – 122]	
[textcomp – 124] [Math – 128] [AMS fonts – 133]	
14.4 Interword Spacing	139
14.5 Additional Kerning	142
15 Auxiliary File for Micro Fine Tuning	143
A Change History	145
B Index	151

List of Tables

1 Predefined font sets	10
2 Fonts with tailored protrusion settings	16
3 Order for matching font attributes	56

1 Micro-Typography with pdf \TeX

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

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

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

Both these features have been lacking a simple \LaTeX user interface for quite some time. Then, the pdfcprot package was released [pdfcprot], which allowed \LaTeX users to employ character protrusion without having to mess much with the internals.

Font expansion, however, was still most difficult to utilize, since it required that the font metrics are available in all levels of expansion. Therefore, anybody who wanted to use this feature had to create multiple instances of the fonts in advance. Shell scripts to partly relieve the user from this burden were available – however, it remained a cumbersome task. Furthermore, all fonts were still being physically created, thus wasting compilation time and disk space.

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

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

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

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

2 Invoking the Package

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

This will be sufficient in most cases, and if you are not interested in fine-tuning the micro-typographic appearance of your document (which would seem unlikely, since using this package is proof of your interest in typographic issues), you may actually skip the rest of this document.

3 Options

Like many other L^AT_EX packages, the microtype package accepts options in the well known key=value syntax. In the following, you'll find a description of all **keys** and their possible values ('true' may be omitted; multiple values, where allowed, must be enclosed in braces; the default value is shown on the right, preceded by an asterisk if it is contingent on the pdfT_EX version).

3.1 Micro-Typographic Options

protrusion true, false, compatibility, nocompatibility, *(font set name)* *true

expansion These are the main options to control the level of micro-typographic refinement, which the fonts in your document should gain. By default, the package is moderately greedy: Character protrusion will be enabled, font expansion will only be disabled in circumstances where pdfT_EX cannot expand the fonts automatically, that is, if it is either too old (versions before 1.20) or if the output mode is DVI (see section 3.4).

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

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

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

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

```
\usepackage{microtype}
```

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

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

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

3.2 Options for Character Protrusion

factor *<integer>* 1000

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

unit character, *<dimension>* character

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

3.3 Options for Font Expansion

auto true, false *true

As noted in chapter 1, the expanded versions of the fonts may be calculated automatically. This option is true by default provided that pdfTeX's version is found to be 1.20 or higher and the output mode is PDF; otherwise, it will be disabled. If auto is set to false, the fonts for all expansion steps must exist (with files called *±<expansion value>*, e.g. `cmr12+10`, as described in the [pdfTeX manual](#), p. 20). If expanded instances of the fonts are available, they will be used regardless whether auto is true or not.

Automatic font expansion requires fonts in Type 1 format. Therefore, if you are using the Computer Modern Roman fonts in T1 encoding⁴, 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.

⁴ En passant, it may be noted that Type 1 format and T1 encoding are in no other way related than that both start with a 'T' and end with a '1'.

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

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

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

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

```
\usepackage{microtype}
```

selected *true, false* false

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

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

3.4 Miscellaneous Options

DVIoutput *true, false* false

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

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

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

draft *true, false* false

final If the `draft` option is passed to the package, *all micro-typographic extensions will be disabled*. The `draft` and `final` options may also be inherited from the class options; of course, you can override them in the package options.

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

verbose true, false, errors false

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

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

config *<file name>* microtype

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

3.5 Changing Settings Later

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

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

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

4 Declaring Font Sets

By default, character protrusion will be applied to all text fonts that are being used in the document, and a basic set of fonts will be expanded. You may want to customize the set of fonts that should get the benefit of micro-typographic treatment. This can be achieved by specifying attributes of the font that have to be matched for them to be taken into account.

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

`\DeclareMicrotypeSet*` This command declares a new set of fonts to which the micro-typographic extensions should be applied. The optional argument may contain a comma-separated list of features to which this set should be restricted. The set can subsequently be activated by calling:

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

The starred version of the command declares *and* activates the font set at the same time.

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

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

If you now call

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

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

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

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

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

If a value is followed by an asterisk (like 'rm*' and 'sf*' in the example above), it does not designate an NFSS code, but will expand to the current `\<value>default`, e.g. `\rmdefault`.⁶ For example, if you want to include the bold font, too, you should say 'bf*' instead of 'b' (`\bfdefault` for Computer Modern is 'bx', while for other fonts, it might be 'b' or even 'sb'). A single asterisk means `\<characteristic>default`, e.g. `\encodingdefault`, respectively `\normalsize` for the size axis.

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

Additionally to this declaration scheme, you can add single fonts to a set using the 'font' key, which expects the concatenation of all font characteristics, separated by forward slashes, i.e., 'font = `<encoding>/<family>/<series>/<shape>/<size>`'. This

⁶ Note that this expansion will take place immediately, so you should make all relevant changes *before* loading the microtype package.

Table 1: Predefined font sets

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

‘*’ = ‘default’

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:

```
\DeclareMicrotypeSet*
[ protrusion ]
{ myset }
{ encoding = T1,
  family   = rm*,
  font     = {T1/tt*/m/n/*,
              T1/tt*/m/it/*} }
```

As you can tell from the example, the asterisk notation is also allowed for the font key. Size selection commands are possible, too, however, ranges are not allowed.

Table 1 lists the six predefined font sets. They may also be activated by passing their name to the feature options expansion and protrusion when loading the package, for example:

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

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

This command activates a font set previously declared by `\DeclareMicrotypeSet`. Using the optional argument, you can limit the application of the set to one or more features.

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

If the package has been loaded without activating any font sets in the package options, the sets declared by this command will be activated (provisionally). By default, the ‘basictext’ font set will be used for font expansion, the ‘alltext’ set for character protrusion.

This command will only have an effect inside the main configuration file (cf. section 5.4). The commands `\DeclareMicrotypeSet` and `\UseMicrotypeSet` may only be used in the preamble or in the main configuration file. Their scope is global to the document. Only one set per feature may be activated.

5 Micro Fine Tuning

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

5.1 Character Protrusion

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

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

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

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

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

The *characters* may be specified either as a single character (‘A’), as a text symbol command (‘`\textquoteleft`’), or as a slot number: three digits for decimal notation, prefixed with ‘#’ for hexadecimal, with ‘o’ for octal (e.g., the ‘fl’ ligature in T1 encoding: 029, #1D, o35). 8-bit characters may be entered directly or in the L^AT_EX 7-bit way of defining them: both ‘Ä’ and ‘\A’ are valid, provided the character is actually included in the encoding(s). You also have the possibility to declare lists of characters that should inherit protrusion or expansion factors (see section 5.3).

The *protrusion factors* designate the amount that a character should be protruded into the left margin (first value) respectively into the right margin (second value). By default, the values are relative to the character widths, so that a value of 1000 means that the character should be shifted fully into the margin, while, for example, with a value of 50 it would be protruded by 5% of its width. Negative values are admitted, as well as numbers larger than 1000 (but effectively not more

than 1em of the font). You can omit either number if the character should not be protruded on that side, but must not drop the separating comma.

The set of fonts to which the settings should apply is declared using the same syntax of $\langle \text{font axis} \rangle = \langle \text{value list} \rangle$ pairs as for the command `\DeclareMicrotypeSet`.

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.⁷ The encoding must always match.

Options:

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

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

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

The following options will affect all loaded lists:

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

```
\SetProtrusion
[ load      = cmr-T1,
  factor    = 700    ]
{ 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

⁷ For the interested, table 3 on page 56 presents the exact order.

measuring of pdfTeX's `\lcode` and `\rcode` primitives). The default behaviour can be restored with `unit=character`.⁸

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

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

5.2 Font Expansion

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

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

The expansion settings consist of $\langle character \rangle = \langle expansion factor \rangle$ pairs.

You may specify one number for each character, which determines the amount that a character may be expanded. The numbers denominate thousandths of the full expansion. For example, if you set the expansion factor for the character ‘O’ to 500, it will only be expanded or shrunk by one half of the amount that the rest of the characters will be expanded or shrunk. While the default value for character protrusion is 0 – that is, if you didn’t specify any characters, none would be protruded –, the default value for expansion is 1000, which means that all characters would be expanded by the same amount.

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

Options:

name, load, preset, context Analogous to `\SetProtrusion`, the optional argument may be used to assign a name to the list, to load another list, to preset all expansion factors, or to determine the context of the list.

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

⁸ 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 ask for a different unit if you are certain that none of the default settings will be used in your document.

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 printed in 10pt:⁹

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

factor This option provides a different method to alter expansion settings for certain fonts, working around another restriction of pdfTeX: It does not allow different expansion limits or steps within one paragraph. The `factor` option influences the expansion factors of all characters (in contrast to the overall stretchability) of the font. For instance, if you want the italic shape to be expanded less, you could declare:

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

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

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

If the `selected` option has been passed to the package (cf. section 3.3), and settings for a font don't exist, font expansion will not be applied to this font at all. Should the extraordinary situation arise that you want to employ `selected`

⁹ 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.

expansion in general but that all characters of a particular font (set) should be expanded or shrunk by the same amount, you would have to declare an empty list for these fonts.

5.3 Character Inheritance

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

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

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

5.4 Configuration Files

The default configuration, consisting of inheritance settings, declarations of font sets and alias fonts, and generic protrusion and expansion settings, will be loaded from the file `microtype.cfg`. You may extend this file with custom settings (or load a different configuration file with the ‘`config`’ option, see section 3.4).

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

This package ships with configuration files for the font families Computer Modern Roman, Palatino, the inescapable Times, Adobe Garamond and Minion¹⁰, for Bitstream Charter and the AMS math fonts. Table 2 lists them all.

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

¹⁰ By courtesy of Harald Harders (h.harders@tu-bs.de).

Table 2: Fonts with tailored protrusion settings

Font family (NFSS code)	Features			
	Series	Shapes	TS1	Math
Generic	m	n, (it, sl, sc) ^a	(•) ^a	
Computer Modern Roman (cmr) ^b	m	n, it, sl, sc	•	•
Bitstream Charter (bch)	m	n, it, (sl) ^c , sc	•	
Adobe Garamond (pad, padx, padj)	m	n, it, (sl) ^c , sc	•	
Adobe Minion (pmnx, pmnj)	m	n, it, (sl) ^c , sc, si	•	
Palatino (ppl, pplx, pplj) ^d	m	n, it, (sl) ^c , sc	(•) ^a	
Times (ptm, ptmx, ptmj) ^e	m	n, it, (sl) ^c , sc	(•) ^a	
AMS math fonts (msa, msb, euf, eus)	m	n		•

^a Incomplete

^b Also used for: Latin Modern (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor)

^c Settings inherited from italic shape

^d Also used for: pxfonts (pxr), qfonts/QuasiPalatino (qpl)

^e Also used for: txfonts (txr), qfonts/QuasiTimes (qtm)

`\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 in the default configuration file, along with others for the virtual fonts provided by the packages `ae`, `zefonts`, `eco` and `hfoldsty`.

`\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.¹¹ This command will load the file `mt-.cfg`.

¹¹ Font package authors might also want to have a look at the hook `\Microtype@Hook`, described in the implementation part, section 13.7.2.

6 Disabling Ligatures

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

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

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

7 Context-sensitive setup

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

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

This command may be used anywhere in the document (also in the preamble) to change the micro-typographic context. For each feature (`protrusion`, `expansion`), one context may be specified. Only settings which have been specified with the corresponding ‘context’ keyword will then be applied. This makes it possible to use different settings for different parts of the document.¹²

8 Hints and Caveats

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

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

¹² This feature is especially useful for the new experimental extensions of pdfTeX: adjustment of interword spacing (glue) and the possibility to specify additional character kerning. The former may improve the appearance of the text even more, the latter allows for instance to insert small spaces before certain characters (e. g., for typesetting in the French tradition) without having to use active characters; also, letterspacing can be implemented in a robust way. Currently, these extensions are only available through patches from <http://pdfTeX.sarovar.org/>. However, if you are adventurous, know how to apply the patches and you are able to compile pdfTeX yourself, you can easily experiment with them, since `microtype` already supports these new extensions. To generate the extended version of the `microtype` package and its documentation, simply remove the comments before ‘\betatrue’ near the beginning of `microtype.ins` and `microtype.dtx`.

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

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

Compatibility. The package should work happily together with all other L^AT_EX packages (except pdfcpot). However, life isn't perfect, so problems are to be expected. Currently, you should be aware of the following issues concerning the loading order of packages:

- All packages that change the default fonts and encodings (e.g. mathpazo, fontenc) should be loaded before microtype, so that variables used in the configuration file (e.g. 'rm*' for \rmdefault) don't expand to a different value than in the body of the document (as explained in section 4).
- When using 8-bit characters in the configuration, inputenc must be loaded first. Unicode input in the configuration is currently not supported.
- The CJK package must be loaded before microtype.¹³

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

¹³ And it might still not work. I simply don't know, since I know nothing about CJK. Feedback on the interaction of both packages – positive or negative – would be appreciated.

added to the document's preamble, would serve the same purpose:¹⁴

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

Possible error messages and how to get rid of them:

- font _____ cannot be expanded (not an included Type1 font)
Font expansion can only be applied if the font is actually embedded in the PDF file. If you receive the above error message, your \TeX system is not set up to embed (or 'download') the base PostScript fonts (e. g. Times, Helvetica, Courier). In most \TeX distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to true. Otherwise, consult the local guide of your \TeX system.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf_mem_size)=65536].
When applying micro-typographic enhancement to a large number of fonts, you may be running out of pdf \TeX memory. You can increase it by setting `pdf_mem_size` to a larger value (maximum 524 288). For $\text{te}\TeX$ -based systems, change the settings in `texmf.cnf`, for MiK \TeX , in the file `miktex.ini`. Beginning with version 1.30 of pdf \TeX , the memory will grow dynamically, so that this problem can no longer arise.

9 Contributions

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

If you have created a configuration file for another font, or if you have any suggestions for enhancements in the default configuration files, I would gratefully accept them: w.m.l@gmx.net.¹⁵

10 Acknowledgments

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

¹⁴ 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.

¹⁵ Should you have lots of pdfcprot configuration files lying around, I can also provide you with a \TeX conversion script. Just ask me.

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.

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

Additionally, the following people have reported bugs or helped otherwise (in chronological order): *Ulrich Dirr*, *Tom Kink*, *Herb Schulz*, *Michael Hoppe*, *Gary L. Gray*, *Georg Verweyen*, *Christoph Bier*, *Peter Muthesius*, *Bernard Gaulle*, *Adam Kucharczyk*, *Mark Rossi*, *Stephan Hennig*, *Michael Zedler*, and *Herbert Voss*.

11 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, *The pdfT_EX user manual*, August 21, 2005. (Available from CTAN at [/systems/pdftex/manual](#); latest version at <http://sarovar.org/projects/pdftex/>)

L^AT_EX3 Project Team, *L^AT_EX2_ε font selection*, February 10, 2004. (Available from CTAN at [/macros/latex/doc/fntguide.pdf](#))

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

12 Short History

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

1.9 (28.10.2005)

New command `\DisableLigatures` to disable ligatures of fonts (requires pdfT_EX version 1.30 or later; see section 6)

New command `\microtypecontext` to change the configuration context; new key ‘context’ for the configuration commands (see section 7)

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

New key ‘preset’ to set all characters to the specified value before loading the lists
 Value ‘relative’ renamed to ‘character’ for ‘unit’ keys
 Support for the Polish OT4 encoding (protrusion, expansion, inheritance)
 Support for the Vietnamese T5 encoding (protrusion, expansion, inheritance)
 ‘DVloutput’ will work with \TeX Live 2004

1.8 (23. 6. 2005)

If font substitution has occurred, the settings for the substitute will be used instead of those for the selected font
 New command `\DeclareMicrotypeSetDefault` to declare the default font sets (see section 4)
 New option ‘config’ to load a different configuration file (see section 3.4)
 New option ‘unit’ to measure protrusion factors relative to a dimension instead of the character width (see section 5.1)
 Renamed commands from `\..MicroType..` to `\..Microtype..`
 Protrusion settings for AMS math fonts
 Protrusion settings for Times in LY1 encoding completed
 The ‘allmath’ font set also includes U encoding
 8-bit characters in the configuration files finally work as advertised, even if made active by the csquotes package
 When using the ledmac package, character protrusion will work for the first time ever (requires pdf \TeX version 1.30 or later)

1.7 (23. 3. 2005)

Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings (see sections 4 and 5)
 Always take font size into account when trying to find protrusion resp. expansion settings for a given font (see section 5)
 New command `\LoadMicrotypeFile` to load a font configuration file manually (see section 5.4)
 Hook `\Microtype@Hook` for font package authors (see section 13.7.2)
 New option ‘verbose=errors’ to turn all warnings into errors
 Disable expansion inside `\showhyphens`
 Warning when running in draft mode

1.6a (2. 2. 2005)

Compatibility with the frenchpro package

1.6 (24. 1. 2005)

New option ‘factor’ to influence protrusion resp. expansion of all characters of a font or font set (see sections 3.2 and 5)
 When pdf \TeX is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled (see section 3.1)
 Protrusion settings of digits improved
 Use e- \TeX extensions, if available

- 1.5 (15. 12. 2004)
 - When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled (see section 3.1)
 - New option ‘selected’ to enable selected expansion (see sections 3.3 and 5.2); default is: false
 - New default for expansion option ‘step’: 4 (min(stretch,shrink)/5) (see section 3.3)
 - Protrusion settings for Bitstream Charter
 - Compatibility with Turkish babel
- 1.4b (26. 11. 2004)
 - `\UseMicrotypeSet` requires the set to be declared (see section 4)
 - Internal optimization
- 1.4 (12. 11. 2004)
 - Set up fonts independently from \LaTeX font loading (therefore, no risk of overlooking fonts anymore, and the package may be loaded at any time)
 - `\microtypesetup` now sets the correct level of protrusion (see chapter 3.5)
 - New option: ‘final’
- 1.3 (27. 10. 2004)
 - Compatibility with the german and ngerman packages
- 1.2 (3. 10. 2004)
 - New font sets: ‘allmath’ and ‘basicmath’ (see section 4 and table 1)
 - Protrusion settings for Computer Modern Roman math symbols
 - Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond
 - If an alias font name is specified, it will be used as an alternative, not as a replacement (see section 5.4)
 - More tests for sanity of settings and whether all fonts will be set up
 - More robust parsing of sizes in font sets
- 1.1 (21. 9. 2004)
 - Protrusion settings for Adobe Minion, contributed by Harald Harders
 - New command: `\DeclareCharacterInheritance` (see section 5.3)
 - Characters may also be specified as octal or hexadecimal numbers (see section 5)
 - Configuration file names in lowercase (see section 5.4)
- 1.0 (11. 9. 2004)
 - First CTAN release

13 Implementation

The docstrip modules in this file are:

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

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

debug: Code for additional output in the log file.
Used for – surprise! – debugging purposes.

config: Surrounds all configuration modules.

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

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

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

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

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

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

pmn: Settings for Adobe Minion (mt-pmn.cfg).
Contributed by Harald Harders.

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

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

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

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

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

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

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

And now for something completely different.

1 *<package>*

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

```

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

```

```

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

```

This command also has a starred version.

```

20 \def\DeclareMicrotypeSet{%
21   \ifstar
22     {\ifnextchar[\MT@DeclareSet{\MT@DeclareSet []}}%
23     {\ifnextchar[\MT@DeclareSet{\MT@DeclareSet []}}%
24 }
25 \def\MT@DeclareSet[#1]#2#3{}

```

Set declarations are only allowed in the preamble (resp. the main configuration file). All the other commands, on the other hand, must be allowed in the document, too, since they may be called inside font configuration files.

```

26 \onlypreamble{\DeclareMicrotypeSet}
27 \onlypreamble{\UseMicrotypeSet}
28 \onlypreamble{\DisableLigatures}

```

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

```

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

```

`\MT@error` Communicate.

```

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

```

Debug. Cases for `\tracingmicrotype`:

- 0: almost none
- 1: + sets & lists
- 2: + heirs
- 3: + slots

4: + factors

```

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

```

13.1 Requirements

\MT@pdftex@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@pdftex@no will be used throughout the package to respectively do the right thing.

Currently, there are 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 (≥ 0.14h)
- 4: + automatic font expansion; default \efcode = 1000 (≥ 1.20)
- 5: + \left|right|marginkern; \pdfnoligatures (≥ 1.30)
- 6: + \left|right|knbs|stbs|shbs|knbc|knac|code (≥ 1.3x)

```

62 \let\MT@pdftex@no\z@

```

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.

```

63 \ifx\normalpdftexversion\undefined \else
64 \let\pdftexversion\normalpdftexversion
65 \let\pdftexrevision\normalpdftexrevision
66 \let\pdfoutput\normalpdfoutput
67 \fi

```

Old packages might have defined \pdftexversion to \relax.

```

68 \ifx\pdftexversion\undefined \else
69 \ifx\pdftexversion\relax \else
70 <debug>\MT@dinfo@n1{0}{running pdftex \the\pdftexversion(\pdftexrevision)}
71 <beta> \def\MT@pdftex@no{5}
72 <beta>
73 \ifx\knbccode\undefined
74 \def\MT@pdftex@no{5}
75 \else
76 \def\MT@pdftex@no{6}
77 \fi
78 </beta>
79 \ifnum\pdftexversion < 130
80 \def\MT@pdftex@no{4}
81 \ifnum\pdftexversion < 120

```

```

82      \let\MT@pdftex@no\thr@
83      \ifnum\pdftexversion = 14
84        \ifnum \expandafter'\pdftexrevision < 'h
85          \let\MT@pdftex@no\tw@
86          \ifnum \expandafter'\pdftexrevision < 'f
87            \let\MT@pdftex@no@ne
88          \fi
89        \fi
90      \else
91        \ifnum\pdftexversion < 14
92          \let\MT@pdftex@no@ne
93        \fi
94      \fi
95    \fi
96  \fi
97 \fi
98 \fi
99 (debug) \MT@dinfo@n1{0}{pdftex no: \number\MT@pdftex@no}

```

If we are not using pdf_{TEX} or in case it is too old, we disable everything and exit here.

```

100 \ifnum\MT@pdftex@no<\tw@
101   \AtEndOfPackage{\let\@unprocessedoptions\relax}%
102   \let\CurrentOption\@empty
103   \MT@warning@n1{%
104     \ifcase\MT@pdftex@no
105       You don't seem to be using pdftex.\MessageBreak
106     \or
107       You are using a pdftex version older than 0.14f.\MessageBreak
108       microtype won't work with such antiquated versions.\MessageBreak
109       Please install a newer version of pdftex.\MessageBreak
110     \fi
111     All micro-typographic features will be disabled}
112   \expandafter
113   \endinput
114 \fi

```

Still there? Then we can begin:

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

```

115 \def\MT@catcodes{%
116   \catcode'\^7 %
117   \@makeother\-%
118   \@makeother\=%
119   \@makeother\*%
120   \@makeother\,%
121   \@makeother\/%
122   \@makeother\'%
123   \@makeother\'%
124   \@makeother\"%
125   \@makeother\|%
126 }

```

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

```

127 \def\MT@restore@catcodes#1{%
128   \ifx\relax#1\else

```

```

129 \noexpand\catcode'\noexpand#1\the\catcode'#1\relax
130 \expandafter\MT@restore@catcodes
131 \fi
132 }
133 \edef\MT@restore@catcodes{%
134 \MT@restore@catcodes\^-\=\*\,\/\'\'\|\relax
135 }
136 \MT@catcodes
137 \AtEndOfPackage{\MT@restore@catcodes}

```

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

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

\mt@toks We need a token register.

```
139 \newtoks\mt@toks
```

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

```

\ifMT@expansion 140 \newif\ifMT@protrusion
\ifMT@auto      141 \newif\ifMT@expansion
142 \newif\ifMT@auto
\ifMT@selected  143 \newif\ifMT@selected
\ifMT@spacing   144 \beta
145 \newif\ifMT@spacing
\ifMT@kerning   146 \newif\ifMT@kerning
147 \beta
\ifMT@DVOutput  148 \newif\ifMT@noligatures
149 \newif\ifMT@DVOutput
\ifMT@babel     150 \newif\ifMT@draft
151 \beta \newif\ifMT@babel

```

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

```

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

```

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

```

163 \let\MT@sp@unit\m@ne
164 \def\MT@kn@unit{1em}
\MT@letterspacing 165 \let\MT@letterspacing\m@ne
166 \beta

```

\MT@pr@min Minimum and maximum values allowed by pdf_T_EX.

```

\MT@pr@max 167 \def\MT@pr@min{-\@m}
168 \let\MT@pr@max\@m
\MT@ex@min 169 \let\MT@ex@min\z@
170 \let\MT@ex@max\@m
\MT@sp@min 171 \beta
172 \def\MT@sp@min{-\@m}
\MT@sp@max 173 \let\MT@sp@max\@m
174 \def\MT@kn@min{-\@m}
\MT@kn@min 175 \let\MT@kn@max\@m
176 \let\MT@kn@max\@m

```

```

176 /beta
\MT@factor@default Default values for expansion.
\MT@stretch@default 177 \def\MT@factor@default{1000 }
\MT@shrink@default 178 \def\MT@stretch@default{20 }
\MT@step@default 179 \def\MT@shrink@default{20 }
180 \def\MT@step@default{4 }
\MT@letterspacing@default Default value for letterspacing (in thousandths of 1em).
181 /beta \def\MT@letterspacing@default{100 }

```

13.2 Compatibility

For the record, the following L^AT_EX commands will be modified by microtype:

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

```

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

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

Peter Wilson was so kind to provide the \l@dunhbox@line hook in ledmac to
allow for protrusion. \leftmarginkern and \rightmarginkern are new primitives
of pdfTEX 1.21b (aka 1.30.0).
191 \def\MT@ledmac@setup{%
192   \ifMT@protrusion
193     \ifnum\MT@pdftex@no > 4
194       \MT@ifdefinedc\l@dunhbox@line{%
195         \MT@info{n1}{Patching ledmac to enable character protrusion}%
196         \newdimen\MT@led@kern
197         \let\MT@led@unhbox@line\l@dunhbox@line
198         \renewcommand*{\l@dunhbox@line}[1]{%
199           \ifhbox##1%
200             \MT@led@kern=\rightmarginkern##1%
201             \kern\leftmarginkern##1%
202             \MT@led@unhbox@line##1%
203             \kern\MT@led@kern

```

```

204     \fi
205   }%
206 }{%
207   \MT@warning@nl{%
208     Character protrusion in paragraphs with line\MessageBreak
209     numbering will only work if you update ledmac}%
210 }%
211 \else
212   \MT@warning@nl{%
213     The pdftex version you are using does not allow\MessageBreak
214     character protrusion in paragraphs with line\MessageBreak
215     numbering by the 'ledmac' package.\MessageBreak
216     Upgrade pdftex to version 1.30 or later}%
217   \fi
218 \fi
219 }

```

`\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.

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

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

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

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

```

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

```

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

```

225 \ifpackageloaded{csquotes}{%
226   \@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax
227 }\relax

```

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

```

228 \ifpackageloaded{hyperref}{%
229   \chardef\%\%
230   \chardef\#\#
231 }\relax
232 }

```

Check again at the end of the preamble.

```
233 \AtBeginDocument{%
```

```

234 \ifpackageloaded{pdfcprot}\MT@pdfcprot@error\relax
235 \ifpackageloaded{ledmac}\MT@ledmac@setup\relax

```

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

```

236 \let\MT@setupfont@hook\empty
237 \ifpackagewith{babel}{spanish}{%
238   \g@addto@macro\MT@setupfont@hook{%
239     \MT@ifdefined@c\percentsign
240     {\let\%\percentsign}\relax}%
241   }\relax
242 \ifpackageloaded{csquotes}{%
243   \ifpackageafter{csquotes}{2005/05/11}{%
244     \g@addto@macro\MT@setupfont@hook\@disablequotes
245   }{%
246     \MT@warning@n1{%
247       Should you receive warnings about unknown slot\MessageBreak
248       numbers, try upgrading the 'csquotes' package}%
249   }%
250   }\relax
251 \ifpackageloaded{hyperref}{%
252   \g@addto@macro\MT@setupfont@hook{%
253     \chardef\%\%
254     \chardef\#\%
255   }%

```

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

```

256 \pdfstringdefDisableCommands{%
257   \let\pickup@font\MT@orig@pickupfont
258 }beta
259 \let\lstyle\empty
260 \let\textls\@firstofone
261 }beta
262 }%
263 }\relax
264 }

```

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

```

265 \expandafter\ifx\the\font\nullfont\normal font\fi

```

13.3 Auxiliary macros

`\MT@etex@no` Test whether we are using e-TeX. Cases:

0: not running e-TeX

1: running e-TeX

```

266 \let\MT@etex@no\z@
267 \ifx\TeXversion\@undefined \else
268   \ifx\TeXversion\relax \else
269     \ifnum\TeXversion>\z@
270       \let\MT@etex@no\ne
271     \fi
272   \fi
273 \fi

```

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

```

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

```

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

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

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

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

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

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

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

`\MT@exp@two@n` Expand the next two tokens after `<#1>` once and enclose them in braces.
282 `\def\MT@exp@two@n#1#2#3{\expandafter\expandafter\expandafter`
283 `#1\expandafter\expandafter\expandafter`
284 `{\expandafter#2\expandafter}\expandafter{#3}}`

You do not wonder why `\MT@exp@one@c` doesn't exist, do you?
`\MT@hop@fi` We need this a couple of lines down to mask `\ifcsname ... \fi` and `\ifdefined ...`
`\MT@hop@else@fi` `\fi`, if `\ifcsname` and `\ifdefined` aren't defined.
285 `\def\MT@hop@fi#1\fi{\fi#1}`
286 `\def\MT@hop@else@fi#1\else#2\fi{\fi#1}`

`\MT@ifdefined@c` Wrapper for testing whether command resp. `\csname` sequence is defined.
`\MT@ifdefined@n` 287 `\ifcase\MT@etex@no \MT@hop@else@fi{%`
288 `debug\MT@info@n1{0}{not running etex}%`
289 `\def\MT@ifdefined@c#1{%`
290 `\ifx#1\@undefined`
291 `\expandafter\@secondoftwo`
292 `\else`
293 `\expandafter\@firstoftwo`
294 `\fi`
295 `}`
296 `\def\MT@ifdefined@n#1{%`
297 `\begingroup\MT@exp@two@c\endgroup`
298 `\ifx\csname #1\endcsname\relax`
299 `\expandafter\@secondoftwo`
300 `\else`
301 `\expandafter\@firstoftwo`
302 `\fi`
303 `}`
304 `}\else\MT@hop@fi{%`

If we are running e-TeX, we will use its primitives `\ifdefined` and `\ifcsname`, which decreases memory use substantially.
305 `debug\MT@info@n1{0}{running etex}%`
306 `\def\MT@ifdefined@c#1{%`
307 `\ifdefined#1%`

```

308     \expandafter\@firstoftwo
309     \else
310     \expandafter\@secondoftwo
311     \fi
312 }
313 \def\MT@ifdefined#1{%
314     \ifcsname#1\endcsname
315     \expandafter\@firstoftwo
316     \else
317     \expandafter\@secondoftwo
318     \fi
319 }
320 }\fi

```

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

```

321 \begingroup
322 \catcode'\%=12
323 \catcode'\&=14
324 \gdef\MT@ifempty#1{%
325     \if %#1%&
326     \expandafter\@firstoftwo
327     \else
328     \expandafter\@secondoftwo
329     \fi
330 }
331 \endgroup

```

`\MT@ifnumber` Test whether argument is a number [0-9] (using an old trick by Mr. Arseneau).

```

332 \def\MT@ifnumber#1{%
333     \if!\ifnum9<1#1!\else?\fi
334     \expandafter\@firstoftwo
335     \else
336     \expandafter\@secondoftwo
337     \fi
338 }

```

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

```

339 \def\MT@ifdimen#1{%
340     \setbox\z@=\hbox{%
341         \MT@count=1#1\relax
342         \ifnum\MT@count=\@ne
343             \aftergroup\@secondoftwo
344         \else
345             \aftergroup\@firstoftwo
346         \fi}%
347 }

```

`\MT@ifgt` Test whether dimensions are smaller, larger or equal.

```

\MT@iflt 348 \def\MT@ifgt#1#2{%
\MT@ifeq 349     \ifdim #1\p@ > #2\p@
350     \expandafter\@firstoftwo
351     \else
352     \expandafter\@secondoftwo
353     \fi
354 }
355 \def\MT@iflt#1#2{%
356     \ifdim #1\p@ < #2\p@
357     \expandafter\@firstoftwo
358     \else

```



```

359 \expandafter\@secondoftwo
360 \fi
361 }
362 \def\MT@ifreq#1#2{%
363 \ifdim #1\p@ = #2\p@
364 \expandafter\@firstoftwo
365 \else
366 \expandafter\@secondoftwo
367 \fi
368 }

```

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

```

369 \def\MT@ifstreq#1#2{%
370 \edef\x{#1}%
371 \edef\y{#2}%
372 \ifx\x\y
373 \expandafter\@firstoftwo
374 \else
375 \expandafter\@secondoftwo
376 \fi
377 }

```

\MT@xadd Add item to a list.

```

378 \def\MT@xadd#1#2{%
379 \ifx#1\relax
380 \xdef#1{#2}%
381 \else
382 \xdef#1{#1#2}%
383 \fi
384 }

```

\MT@xaddb Add item to the beginning.

```

385 \def\MT@xaddb#1#2{%
386 \ifx#1\relax
387 \xdef#1{#2}%
388 \else
389 \xdef#1{#2#1}%
390 \fi
391 }

```

\MT@map@clist@n Run $\langle\#2\rangle$ on all elements of the comma list $\langle\#1\rangle$. This and the following is modelled after \LaTeX 3 commands.

\MT@map@clist@c

\MT@map@clist@

\MT@clist@function

\MT@clist@break

```

392 \def\MT@map@clist@n#1#2{%
393 \ifx\@empty#1\else
394 \def\MT@clist@function##1{#2}%
395 \expandafter\MT@map@clist@
396 \expandafter#1,\@nil,\@nnil,%
397 \fi
398 }
399 \def\MT@map@clist@c#1{\expandafter\MT@map@clist@n\expandafter{#1}}
400 \def\MT@map@clist@#1,{%
401 \ifx\@nil#1%
402 \MT@clist@break
403 \else
404 \MT@clist@function{#1}%
405 \expandafter\MT@map@clist@
406 \fi
407 }
408 \def\MT@clist@break#1\@nnil,{\fi}

```

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

```
\MT@map@tlist@ 409 \def\MT@map@tlist@n#1#2{%
\MT@map@tlist@c 410   \MT@map@tlist@#2#1\@nnil
\MT@tlist@break 411 }
                  412 \def\MT@map@tlist@c#1#2{%
                  413   \expandafter\MT@map@tlist@
                  414   \expandafter#2#1\@nnil
                  415 }
                  416 \def\MT@map@tlist@#1#2{%
                  417   \ifx\@nnil#2\else
                  418     #1{#2}%
                  419     \expandafter\MT@map@tlist@
                  420     \expandafter#1%
                  421   \fi
                  422 }
                  423 \def\MT@tlist@break#1\@nnil{\fi}
```

`\ifMT@inlist@` Test whether item $\langle\#1\rangle$ is in comma list $\langle\#2\rangle$.

```
\MT@in@clist 424 \newif\ifMT@inlist@
               425 \def\MT@in@clist#1#2{%
               426   \MT@inlist@false
               427   \def\x##1#1,##2\@nnil{%
               428     \ifx\##2\\\else
               429       \MT@inlist@true
               430     \fi
               431   }%
               432   \expandafter\x#2,#1,\@nnil
               433 }
```

`\MT@rem@from@list` Remove item $\langle\#1\rangle$ from comma list $\langle\#2\rangle$.

```
434 \def\MT@rem@from@list#1#2{%
435   \def\x##1#1,##2\@nnil{%
436     \ifx\##2\\\else
437       \def\x###1,#1,###2\@nnil{%
438         \gdef#2{##1###1}%
439       }%
440       \x##2,#1,\@nnil
441     \fi
442   }%
443   \expandafter\x#2,#1,\@nnil
444 }
```

`\MT@in@tlist` Test whether item is in token list.

```
445 \def\MT@in@tlist#1#2{%
446   \MT@inlist@false
447   \def\x{#1}%
448   \MT@map@tlist@c#2\MT@in@tlist@
449 }
450 \def\MT@in@tlist@#1{%
451   \edef\y{#1}%
452   \ifx\x\y
453     \MT@inlist@true
454     \expandafter\MT@tlist@break
455   \fi
456 }
```

`\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@in@rlist@@
\MT@size@name
```

```

\MT@size@name
457 \def\MT@in@rlist#1{%
458   \MT@inlist@false
459   \MT@map@tlist@c#1\MT@in@rlist@
460 }
461 \def\MT@in@rlist@#1{%
462   \expandafter\MT@in@rlist@@#1%
463 }
464 \def\MT@in@rlist@@#1#2#3{%
465   \MT@ifeq{#2}\m@ne{%
466     \MT@ifeq{#1}\MT@size
467     \MT@inlist@true
468     \relax
469   }{%
470     \MT@iflt\MT@size{#1}\relax{%
471       \MT@iflt\MT@size{#2}%
472       \MT@inlist@true
473       \relax
474     }%
475   }%
476   \ifMT@inlist@
477   \def\MT@size@name{#3}%
478   \expandafter\MT@tlist@break
479   \fi
480 }

```

`\MT@loop` This is the same as L^AT_EX's `\loop`, which we mustn't use, since this could confuse an outer `\loop` in the document.

```

\MT@iterate
\MT@repeat 481 \def\MT@loop#1\MT@repeat{%
482   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
483   \MT@iterate
484   \let\MT@iterate\relax
485 }
486 \let\MT@repeat\fi

```

`\MT@while@num` Sweetness.

```
487 \def\MT@while@num#1#2{\MT@loop #2\ifnum#1\MT@repeat}
```

`\MT@count` Increment macro `<#1>` by one. Saves using up too many counters.

```

\MT@increment 488 \newcount\MT@count
489 \ifcase\MT@etex@no
490   \def\MT@increment#1{%
491     \MT@count=#1\relax
492     \advance\MT@count \@ne
493     \edef#1{\number\MT@count}%
494   }
495 \else

```

The e-_T_EX way is slightly faster.

```

496   \def\MT@increment#1{%
497     \edef#1{\number\numexpr #1 + 1\relax}%
498   }
499 \fi

```

`\MT@scale` Multiply and divide a counter. If we are using e-_T_EX, 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.

```

500 \ifcase\MT@etex@no
501   \def\MT@scale#1#2#3{%
502     \multiply #1 #2\relax
503     \ifnum #3 = \z@ \else
504       \divide #1 #3\relax
505     \fi
506   }
507 \else
508   \def\MT@scale#1#2#3{%
509     \ifnum #3 = \z@
510       #1=\numexpr #1 * #2\relax
511     \else
512       #1=\numexpr #1 * #2 / #3\relax
513     \fi
514   }
515 \fi

```

`\MT@remove@spaces` Remove spaces around $\langle\#1\rangle$ (`\KV@sp@def` is from `keyval`). It will be neutralized in `\MT@begin@catcodes`.

```

516 \def\MT@remove@spaces#1{\expandafter\KV@sp@def\expandafter#1\expandafter{#1}}

```

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

```

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

```

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

```

\MT@abbr@ex
\MT@abbr@pr@c 518 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 519 \def\MT@abbr@ex{expansion}
520 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@pr@inh 521 \def\MT@abbr@ex@c{expansion codes}
522 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@ex@inh 523 \def\MT@abbr@ex@inh{expansion inheritance}
\MT@abbr@nl 524 \def\MT@abbr@nl{noligatures}
\MT@abbr@sp 525 \beta
\MT@abbr@sp@c 526 \def\MT@abbr@sp{spacing}
527 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@sp@inh 528 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@kn 529 \def\MT@abbr@kn{kerning}
\MT@abbr@kn@c 530 \def\MT@abbr@kn@c{kerning codes}
\MT@abbr@kn@inh 531 \def\MT@abbr@kn@inh{kerning inheritance}
532 /\beta

```

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

```

\MT@rbba@expansion 533 \def\MT@rbba@protrusion{pr}
534 \def\MT@rbba@expansion{ex}
\MT@rbba@spacing 535 \beta
\MT@rbba@kerning 536 \def\MT@rbba@spacing{sp}
537 \def\MT@rbba@kerning{kn}
538 /\beta

```

13.4 Setting up a font

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

```

539 \def\MT@setupfont{%
540   \ifx\MT@vinfo\MT@info@nl

```

```
541 \MT@info{Setting up font '\MT@exp@string\MT@font'}\fi
```

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

```
542 \MT@setupfont@hook
```

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

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

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

```
544 \MT@exp@one@n\MT@find@file\MT@family
545 \ifx\MT@familyalias\@empty \else
546 \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).

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

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

Protrusion has to be set up first, says Thành!

```
548 \MT@protrusion
549 \MT@expansion
```

Interword spacing and kerning.

```
550 <*\beta>
551 \MT@spacing
552 \MT@kerning
553 </\beta>
```

Disable ligatures?

```
554 \MT@noligatures
555 }
```

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

```
\MT@encoding 556 \def\MT@split@name#1/#2/#3/#4/#5\@nil{%
\MT@family    557 \def\MT@encoding{#1}%
\MT@family    558 \def\MT@family{#2}%
\MT@series    559 \def\MT@series{#3}%
\MT@shape     560 \def\MT@shape{#4}%
\MT@size      561 \def\MT@size{#5}%
```

`\MT@familyalias` Alias family?

```
562 \MT@ifdefined@n{\MT@MT@family @alias}%
563 {\MT@let@cn\MT@familyalias{\MT@MT@family @alias}}%
564 {\let\MT@familyalias\@empty}%
565 }
```

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

`\MT@feat`

```
\MT@maybe@do 566 \newif\ifMT@do
567 \def\MT@maybe@do#1{%
```

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

```
568 \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).

```

569 \MT@dotrue
570 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
571 \MT@ifdefined@n{MT@checklist@##1}%
572 {\csname MT@checklist@##1\endcsname}%
573 {\MT@checklist@{##1}}%
574 {#1}%
575 }%
576 \else
577 \MT@dofalse
578 \fi
579 \ifMT@do

```

`\MT@feat` stores the current feature.

```

580 \def\MT@feat{#1}%
581 \csname MT@set@#1@codes\endcsname
582 \else
583 \MT@vinfo{... No \nameuse{MT@abbr@#1}}%
584 \fi
585 }

```

`\MT@checklist@` The generic test.

```

586 \def\MT@checklist@#1#2{%
587 \edef\@tempa{\csname MT@#2@setname\endcsname}%
588 \MT@ifdefined@n{MT@#2list@#1@\@tempa}{%

```

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

```

589 \expandafter\expandafter\expandafter
590 \MT@in@clist\expandafter\expandafter\expandafter
591 {\csname MT@#1\expandafter\endcsname\expandafter}%
592 \csname MT@#2list@#1@\@tempa\endcsname
593 \ifMT@inlist@
594 <debug>\MT@dinfo@n{1}{\nameuse{MT@abbr@#2}: #1 '\nameuse{MT@#1}' in list}%
595 \MT@dotrue
596 \else
597 <debug>\MT@dinfo@n{1}{\nameuse{MT@abbr@#2}: #1 '\nameuse{MT@#1}' not in list}%
598 \MT@dofalse
599 \expandafter\MT@clist@break
600 \fi
601 }{%

```

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

```

602 <debug>\MT@dinfo@n{1}{\nameuse{MT@abbr@#2}: #1 list empty}%
603 }%
604 }

```

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

```

605 \def\MT@checklist@font#1{%
606 \edef\@tempa{\csname MT@#1@setname\endcsname}%
607 \MT@ifdefined@n{MT@#1list@font@\@tempa}{%
608 \MT@exp@two@n\MT@in@clist
609 \MT@font{\csname MT@#1list@font@\@tempa\endcsname}%
610 \ifMT@inlist@
611 <debug>\MT@dinfo@n{1}{\nameuse{MT@abbr@#1}: font '\MT@font' in list}%
612 \expandafter\MT@clist@break

```

```

613     \else
614 (debug) \MT@info@n{1}{\@nameuse{MT@abbr@#1}: font '\MT@font' not in list}%
615         \MT@dofalse
616     \fi
617 }{%
618 (debug) \MT@info@n{1}{\@nameuse{MT@abbr@#1}: font list empty}%
619 }%
620 }

```

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

```

621 \def\MT@checklist@family#1{%
622   \edef\@tempa{\csname MT@#1@setname\endcsname}%
623   \MT@ifdefined@{MT@#1list@family@}{\@tempa}{%
624     \MT@exp@two@n\MT@in@clist
625     \MT@family{\csname MT@#1list@family@}{\@tempa\endcsname}%
626     \ifMT@inlist@
627 (debug) \MT@info@n{1}{\@nameuse{MT@abbr@#1}: family '\@nameuse{MT@family}' in list}%
628         \MT@dotrue
629     \else
630 (debug) \MT@info@n{1}{\@nameuse{MT@abbr@#1}: family '\@nameuse{MT@family}' not in list}%
631         \MT@dofalse
632         \ifx\MT@familyalias\@empty \else
633           \MT@exp@two@n\MT@in@clist
634           \MT@familyalias{\csname MT@#1list@family@}{\@tempa\endcsname}%
635           \ifMT@inlist@
636 (debug) \MT@info@n{1}{\@nameuse{MT@abbr@#1}: alias '\MT@familyalias' in list}%
637               \MT@dotrue
638 (debug) \else\MT@info@n{1}{\@nameuse{MT@abbr@#1}: alias '\MT@familyalias' not in list}%
639           \fi
640         \fi
641     \fi
642     \ifMT@do \else
643       \expandafter\MT@clist@break
644     \fi
645   }{%
646 (debug) \MT@info@n{1}{\@nameuse{MT@abbr@#1}: family list empty}%
647 }%
648 }

```

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

```

649 \def\MT@checklist@size#1{%
650   \edef\@tempa{\csname MT@#1@setname\endcsname}%
651   \MT@ifdefined@{MT@#1list@size@}{\@tempa}{%
652     \expandafter\MT@in@rlist
653     \csname MT@#1list@size@}{\@tempa\endcsname
654     \ifMT@inlist@
655 (debug) \MT@info@n{1}{\@nameuse{MT@abbr@#1}: size '\MT@size' in list}%
656         \MT@dotrue
657     \else
658 (debug) \MT@info@n{1}{\@nameuse{MT@abbr@#1}: size '\MT@size' not in list}%
659         \MT@dofalse
660         \expandafter\MT@clist@break
661     \fi
662   }{%
663 (debug) \MT@info@n{1}{\@nameuse{MT@abbr@#1}: size list empty}%
664 }%
665 }

```

13.4.1 Protrusion

`\MT@protrusion` Set up for protrusion?

```
666 \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.

```
667 \def\MT@set@pr@codes{%
668   \MT@reset@pr@codes

Check whether and if, which list should be applied to the current font.
669   \MT@if@list@exists{%
670     \MT@get@dimen@six
671     \MT@get@opt

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

Load additional lists?
673     \MT@load@list{\MT@pr@c@name}%

Load the main list.
674     \edef\MT@curr@list@name{protrusion list '\MT@pr@c@name'}%
675     \MT@let@cn\@tempc{\MT@pr@c@\MT@pr@c@name}%
676     \expandafter\MT@pr@do\@tempc,\relax,%
677   }\relax
678 }
```

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

```
679 \def\MT@set@all@pr#1#2{%
680   debug \MT@info@nl{3}{-- lp/rp: setting all to \number#1/\number#2}%
681   \@tempcnta=\z@
682   \MT@while@num{\@tempcnta < \ccclvi}{%
683     \lcode\MT@font\@tempcnta=#1\relax
684     \rcode\MT@font\@tempcnta=#2\relax
685     \advance\@tempcnta \@ne
686   }%
687 }
```

`\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.

```
688 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
689 \let\MT@reset@pr@codes\relax
```

`\MT@gobble@settings` If `\fontdimen 6` is zero, character protrusion won't work, and we can skip the settings (for example, the `dsfont` fonts don't specify this dimension; this is probably a bug).

```
690 \def\MT@get@dimen@six{%
691   \ifnum\fontdimen6\MT@font=\z@
692     \MT@warning@nl{%
693       Font '\MT@exp@string\MT@font' does not specify its\MessageBreak
694       \@backslashchar fontdimen 6 (width of an 'em')! Therefore,\MessageBreak
695       \@nameuse{MT@abbr@\MT@feat} will not work with this font}%
696     \expandafter\MT@gobble@settings
697   \else
698     \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
699   \fi
```



```

700 }
701 \def\MT@gobble@settings#1\@tempc,\relax,{}

\MT@pr@do Split up the values and set \lcode and \rcode.
702 \def\MT@pr@do#1,{%
703   \ifx\relax#1\@empty\else
704     \MT@pr@split #1=\relax
705     \expandafter\MT@pr@do
706   \fi
707 }

\MT@pr@split The keyval package would remove spaces here, which we needn't do since
\SetProtrusion ignores spaces in the protrusion list anyway.
708 \def\MT@pr@split#1=#2=#3\relax{%
709   \def\@tempa{#1}%
710   \ifx\@tempa\@empty \else
711     \MT@get@slot
712     \ifnum\MT@char > \m@ne
713       \MT@get@char@unit
714       \MT@pr@split@val#2\relax
715     \fi
716   \fi
717 }

\MT@pr@split@val
718 \def\MT@pr@split@val#1,#2\relax{%
719   \def\@tempb{#1}%
720   \MT@ifempty\@tempb\relax{%
721     \MT@scale@to@em
722     \lcode\MT@font\MT@char=\@tempcntb
723     (debug)\MT@info@nl{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char}%
724   }%
725   \def\@tempb{#2}%
726   \MT@ifempty\@tempb\relax{%
727     \MT@scale@to@em
728     \rcode\MT@font\MT@char=\@tempcntb
729     (debug)\MT@info@nl{4}{;;; rp (\MT@char): \number\rcode\MT@font\MT@char}%
730   }%

Now we can set the values for the inheriting characters. Their slot numbers are
saved in the macro \MT@inh@<list name>@<slot number>@.
731 \MT@ifdefinedc\MT@pr@inh@name{%
732   \MT@ifdefinedn\MT@inh@\MT@pr@inh@name @\MT@char @{%
733     \expandafter\MT@map@tlistc
734     \csname MT@inh@\MT@pr@inh@name @\MT@char @\endcsname
735     \MT@set@pr@heirs
736   } \relax
737 } \relax
738 }

\MT@scale@to@em Since pdfTeX version 0.14h, we have to adjust the protrusion factors (i. e., convert
numbers from thousandths of character width to thousandths of an em of the
font). We have to do this before setting the inheriting characters, so that the latter
inherit the absolute value, not the relative one if they have a differing width (e. g.
the ‘ff’ ligature). Unlike protcode.tex and pdfcprot, we do not calculate with
\lcode resp. \rcode, since this would disallow protrusion factors larger than the
character width (since \[1r]pcode’s limit is 1000). Now, the maximum protrusion

```

is 1em of the font.

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

```
739 \ifnum\MT@pdfTeX@no > \tw@
740   \def\MT@scale@to@em{%
741     \@tempcntb=\MT@count\relax
```

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

```
742   \MT@scale\@tempcntb \@tempb \MT@dimen@six
743   \ifnum\@tempcntb=\z@ \else
744     \MT@scale@factor
745   \fi
746 }
```

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

```
747 \ifcase\MT@etex@no
748   \def\MT@get@charwd{%
749     \setbox\z@=\hbox{\MT@font \char\MT@char}%
750     \MT@count=\wd\z@\relax
751     \ifnum\MT@count=\z@ \MT@warn@missing@char \fi
752   }
753 \else
754   \def\MT@get@charwd{%
755     \MT@count=\number\fontcharwd\MT@font\MT@char\relax
756     \ifnum\MT@count=\z@ \MT@warn@missing@char \fi
757   }
758 \fi
```

No adjustment with versions 0.14f and 0.14g.

```
759 \else
760   \def\MT@scale@to@em{%
761     \MT@count=\@tempb\relax
762     \ifnum\MT@count=\z@ \else
763       \MT@scale@factor
764     \fi
765 }
```

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

```
766 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
767 \fi
```

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

```
768 \def\MT@get@font@dimen#1{%
769   \MT@count=\number\fontdimen#1\MT@font
770 }
```

`\MT@warn@missing@char` Warning for missing characters, or characters with zero width.

```
771 \ifcase\MT@etex@no \MT@hop@else@fi{%
772   \def\MT@warn@missing@char{%
773     \MT@warning@nl{%
774       Character '\the\mt@toks' has a width of 0pt\MessageBreak
775       (it's probably missing) in font '\MT@exp@string\MT@font'.\MessageBreak
776       It cannot be protruded}%
777   }
778 }\else\MT@hop@fi{%
```

```

779 \def\MT@warn@missing@char{%
780 \MT@warning@n1{Character '\the\mt@toks'
781 \iffontchar\MT@font\MT@char has a width of 0pt \else is missing \fi
782 in font\MessageBreak '\MT@exp@string\MT@font'. It cannot be protruded}%
783 }
784 }\fi

```

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

```

785 \def\MT@scale@factor{%
786 \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
787 \expandafter\MT@scale\expandafter
788 \@tempcntb \csname MT@\MT@feat @factor@\endcsname \@m
789 \fi
790 \ifnum\@tempcntb > \csname MT@\MT@feat @max\endcsname\relax
791 \@tempcnta=\csname MT@\MT@feat @max\endcsname
792 \MT@warn@code@too@large
793 \else
794 \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
795 \@tempcnta=\csname MT@\MT@feat @min\endcsname
796 \MT@warn@code@too@large
797 \fi
798 \fi
799 }

```

\MT@warn@code@too@large Type out a warning if a chosen protrusion factor is too large after the conversion. As a special service, we also type out the maximum amount that may be specified in the configuration file.

```

800 \def\MT@warn@code@too@large{%
801 \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
802 \expandafter\MT@scale\expandafter\@tempcnta\expandafter\@m
803 \csname MT@\MT@feat @factor@\endcsname
804 \fi
805 \MT@scale\@tempcnta \MT@dimen@six \MT@count
806 \MT@warning@n1{The \nameuse{MT@abbr@\MT@feat} code \@tempb\space
807 is too large for character\MessageBreak
808 '\the\mt@toks' in \MT@curr@list@name.\MessageBreak
809 Setting it to the maximum of \number\@tempcnta}%
810 \@tempcntb=\@tempcnta
811 }

```

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

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

\MT@pr@factor@ Apply a factor?

```

\MT@sp@factor@ 813 \MT@ifdefined@{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 814 \MT@let@nn{MT@\MT@feat @factor@}
815 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
816 \MT@vinfo{... : Multiplying \nameuse{MT@abbr@\MT@feat} codes by
817 \number\csname MT@\MT@feat @factor@\endcsname/1000}%
818 }{%
819 \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
820 }%

```

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

```

\MT@sp@unit@
\MT@kn@unit@ 821 \MT@ifdefined@{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
822 \MT@let@nn{MT@\MT@feat @unit@}%

```

```

823      {MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @unit}%
824      \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\empty
825      \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
826      relative to character widths}%
827      \else
828      \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\m@ne
829      \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} codes
830      relative to width of space}%
831      \fi
832      \fi
833      }{%
834      \MT@let@nn{MT@MT@feat @unit@}{MT@MT@feat @unit}%
835      }%

```

\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.

```

836      \let\MT@get@char@unit\relax
837      \let\MT@get@space@unit@gobble
838      \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\empty
839      \let\MT@get@char@unit\MT@get@charwd
840      \else
841      \expandafter\ifx\csname MT@MT@feat @unit@\endcsname\m@ne
842      \let\MT@get@space@unit\MT@get@font@dimen
843      \else
844      \expandafter\MT@get@unit\csname MT@MT@feat @unit@\endcsname
845      \fi
846      \fi

```

Preset all characters?

```

847      \MT@ifdefined@{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @preset}{%
848      \csname MT@preset@MT@feat\endcsname
849      }\relax
850      }

```

\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.

```

851      \def\MT@get@unit#1{%
852      \expandafter\MT@get@unit@#1 e!\@nil
853      \ifx\x\@empty\else\let#1\x\fi
854      \@defaultunits\@tempdima#1 pt\relax\@nnil
855      \ifdim\@tempdima=\z@
856      \MT@warning@n1{%
857      Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
858      width. Setting factors of list '\@nameuse{MT@MT@feat @c@name}'\MessageBreak
859      relative to character widths instead}%
860      \let#1\@empty
861      \let\MT@get@char@unit\MT@get@charwd
862      \else
863      \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
864      to \the\@tempdima}%
865      \MT@count=\number\@tempdima\relax
866      \fi
867      }
868      \def\MT@get@unit@#1e#2#3\@nil{%
869      \ifx\#3\\\let\x\@empty \else

```

```

870 \if m#2%
871 \edef\x{#1\fontdimen6\MT@font}%
872 \else
873 \if x#2%
874 \edef\x{#1\fontdimen5\MT@font}%
875 \fi
876 \fi
877 \fi
878 }

```

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

```

879 \def\MT@set@pr@heirs#1{%
880 \lcode\MT@font#1=\lcode\MT@font\MT@char
881 \rcode\MT@font#1=\rcode\MT@font\MT@char
882 <*debug>
883 \MT@info{n}{2}{-- heir of \MT@char: #1}%
884 \MT@info{n}{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
885 \number\rcode\MT@font\MT@char}%
886 </debug>
887 }

```

`\MT@preset@pr`

```

\MT@preset@pr@ 888 \def\MT@preset@pr{%
889 \expandafter\expandafter\expandafter\MT@preset@pr@
890 \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
891 }
892 \def\MT@preset@pr@#1,#2\@nil{%
893 \ifx\MT@pr@unit@\@empty
894 \MT@warning{n}{%
895 Cannot preset characters relative to their widths\MessageBreak
896 for protrusion list '\MT@pr@c@name'. Presetting them\MessageBreak
897 relative to lem instead}%
898 \let\MT@preset@aux\MT@preset@aux@factor
899 \else
900 \let\MT@preset@aux\MT@preset@aux@space
901 \fi
902 \MT@preset@aux{#1}\@tempa
903 \MT@preset@aux{#2}\@tempb
904 \MT@set@all@pr\@tempa\@tempb
905 }

```

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

```

\MT@preset@aux@factor 906 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 907 \@tempcntb=#1\relax
908 \MT@scale@factor
909 \edef#2{\number\@tempcntb}%
910 }
911 \def\MT@preset@aux@space#1#2{%
912 \def\@tempb{#1}%
913 \MT@get@space@unit\tw@
914 \MT@scale@to@em
915 \edef#2{\number\@tempcntb}%
916 }

```

13.4.2 Expansion

`\MT@expansion` Set up for expansion?

```

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

```

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

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

```

918 \def\MT@set@ex@codes@{%
919   \MT@if@list@exists{%
920     \MT@get@ex@opt
921     \MT@reset@ef@codes
922     \MT@get@inh@list
923     \MT@load@list{\MT@ex@cc@name}%
924     \edef\MT@curr@list@name{expansion list '\MT@ex@cc@name'}%
925     \MT@let@cn@tempc{\MT@ex@cc@MT@ex@cc@name}%
926     \expandafter\MT@ex@do\@tempc,\relax,%
927     \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
928   }\relax
929 }
```

`\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.

```

930 \newif\ifMT@nonselected

931 \def\MT@set@ex@codes@n{%
932   \MT@nonselectedtrue
933   \MT@if@list@exists
934   \MT@get@ex@opt
935   {%
936     \let\MT@stretch@\MT@stretch
937     \let\MT@shrink@\MT@shrink
938     \let\MT@step@\MT@step
939     \let\MT@auto@\MT@auto
940     \let\MT@ex@factor@\MT@ex@factor
941   }%
942   \MT@reset@ef@codes
943   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
944   \MT@nonselectedfalse
945 }
```

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

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

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

```

947 \def\MT@set@all@ex#1{%
948   debug \MT@info{n}{3}{-- ex: setting all to \number#1}%
949   \@tempcnta=\z@
950   \MT@while@num{\@tempcnta < \@cc@lvi}{%
951     \efcode\MT@font\@tempcnta=#1\relax
952     \advance\@tempcnta \@ne
953   }%
954 }
955 \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.

```

956 \ifnum\MT@pdf@tex@no < 4
957   \let\MT@reset@ef@codes\MT@reset@ef@codes@
```

```

958 \else
959   \def\MT@reset@ef@codes{%
960     \ifnum\MT@ex@factor@=\@m \else
961       \MT@reset@ef@codes@
962     \fi
963   }
964 \fi

```

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

```

965 \def\MT@ex@do#1,{%
966   \ifx\relax#1\@empty \else
967     \MT@ex@split #1=\relax
968     \expandafter\MT@ex@do
969   \fi
970 }

```

\MT@ex@split

```

971 \def\MT@ex@split#1=#2=#3\relax{%
972   \def\@tempa{#1}%
973   \ifx\@tempa\@empty \else
974     \MT@get@slot
975     \ifnum\MT@char > \m@ne
976       \@tempcntb=#2\relax

```

Take an optional factor into account.

```

977     \ifnum\MT@ex@factor@=\@m \else
978       \MT@scale\@tempcntb \MT@ex@factor@ \@m
979     \fi
980     \ifnum\@tempcntb > \MT@ex@max
981       \MT@warn@ex@too@large\MT@ex@max
982     \else
983       \ifnum\@tempcntb < \MT@ex@min
984         \MT@warn@ex@too@large\MT@ex@min
985       \fi
986     \fi
987     \efcode\MT@font\MT@char=\@tempcntb
988 (debug) \MT@dinfo@n1{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char}%

```

Heirs, heirs, I love thy heirs.

```

989 \MT@ifdefined@c\MT@ex@inh@name{%
990   \MT@ifdefined@n{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
991     \expandafter\MT@map@tlist@c
992     \csname MT@inh@\MT@ex@inh@name @\MT@char @\endcsname
993     \MT@set@ex@heirs
994   }\relax
995 } \relax
996 \fi
997 \fi
998 }

```

\MT@warn@ex@too@large

```

999 \def\MT@warn@ex@too@large#1{%
1000   \MT@warning@n1{Expansion factor
1001     \number\@tempcntb\space too large for character\MessageBreak
1002     '\the\mt@toks' in \MT@curr@list@name.\MessageBreak
1003     Setting it to the maximum of \number#1}%
1004   \@tempcntb=#1\relax
1005 }

```

```

\MT@get@ex@opt Apply different values to this font?
\MT@ex@factor@ 1006 \def\MT@get@ex@opt{%
\MT@stretch@ 1007 \MT@ifdefined@{\MT@ex@c@\MT@ex@c@name @factor}{%
1008 \MT@let@cn\MT@ex@factor@{\MT@ex@c@\MT@ex@c@name @factor}%
\MT@shrink@ 1009 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@step@ 1010 }{%
1011 \let\MT@ex@factor@\MT@ex@factor
\MT@auto@ 1012 }%
1013 \MT@ifdefined@{\MT@ex@c@\MT@ex@c@name @stretch}{%
1014 \MT@let@cn\MT@stretch@{\MT@ex@c@\MT@ex@c@name @stretch}%
1015 \MT@vinfo{... : Setting stretch limit to \number\MT@stretch@}%
1016 }{%
1017 \let\MT@stretch@\MT@stretch
1018 }%
1019 \MT@ifdefined@{\MT@ex@c@\MT@ex@c@name @shrink}{%
1020 \MT@let@cn\MT@shrink@{\MT@ex@c@\MT@ex@c@name @shrink}%
1021 \MT@vinfo{... : Setting shrink limit to \number\MT@shrink@}%
1022 }{%
1023 \let\MT@shrink@\MT@shrink
1024 }%
1025 \MT@ifdefined@{\MT@ex@c@\MT@ex@c@name @step}{%
1026 \MT@let@cn\MT@step@{\MT@ex@c@\MT@ex@c@name @step}%
1027 \MT@vinfo{... : Setting expansion step to \number\MT@step@}%
1028 }{%
1029 \let\MT@step@\MT@step
1030 }%
1031 \MT@ifdefined@{\MT@ex@c@\MT@ex@c@name @auto}{%
1032 \MT@let@cn\MT@auto@{\MT@ex@c@\MT@ex@c@name @auto}%
1033 \def\@tempa{autoexpand}%
1034 \MT@vinfo{... : \ifx\@tempa\MT@auto@ En\else Dis\fi
1035 abling automatic expansion}%
1036 }{%
1037 \let\MT@auto@\MT@auto
1038 }%
1039 }

\MT@set@ex@heirs
1040 \def\MT@set@ex@heirs#1{%
1041 \efcode\MT@font#1=\efcode\MT@font\MT@char
1042 <*debug>
1043 \MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
1044 \MT@dinfo@nl{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1045 </debug>
1046 }

\MT@preset@ex
1047 \def\MT@preset@ex{%
1048 \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
1049 \MT@scale@factor
1050 \MT@set@all@ex\@tempcntb
1051 }
1052 <*beta>

```

13.4.3 Interword Space (Glue)

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

```
1053 \ifnum\MT@pdfTeX@no > 5
```



```

1054 \def\MT@spacing{\MT@maybe@do{sp}}
1055 \else
1056 \let\MT@spacing\relax
1057 \fi

\MT@set@sp@codes This is all the same.
1058 \def\MT@set@sp@codes{%
1059 \MT@reset@sp@codes
1060 \MT@if@list@exists{%
1061 \MT@get@dimen@six
1062 \MT@get@opt
1063 \MT@get@inh@list
1064 \MT@load@list{\MT@sp@cc@name}%
1065 \edef\MT@curr@list@name{spacing list '\MT@sp@cc@name'}%
1066 \MT@let@cn\@tempc{\MT@sp@cc@\MT@sp@cc@name}%
1067 \expandafter\MT@sp@do\@tempc,\relax,%
1068 }\relax
1069 }

\MT@sp@do
1070 \def\MT@sp@do#1,{%
1071 \ifx\relax#1\@empty \else
1072 \MT@sp@split #1=\relax
1073 \expandafter\MT@sp@do
1074 \fi
1075 }

\MT@sp@split
1076 \def\MT@sp@split#1=#2=#3\relax{%
1077 \def\@tempa{#1}%
1078 \ifx\@tempa\@empty \else
1079 \MT@get@slot
1080 \ifnum\MT@char > \m@ne
1081 \MT@get@char@unit
1082 \MT@sp@split@val#2\relax
1083 \fi
1084 \fi
1085 }

\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).
1086 \def\MT@sp@split@val#1,#2,#3\relax{%
1087 \def\@tempb{#1}%
1088 \MT@ifempty\@tempb\relax{%
1089 \MT@get@space@unit\tw@
1090 \MT@scale@to@em
1091 \knbscode\MT@font\MT@char=\@tempcntb
1092 debug\MT@info@n1{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char}%
1093 }%
1094 \def\@tempb{#2}%
1095 \MT@ifempty\@tempb\relax{%
1096 \MT@get@space@unit\thr@@
1097 \MT@scale@to@em
1098 \stbscode\MT@font\MT@char=\@tempcntb
1099 debug\MT@info@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char}%
1100 }%
1101 \def\@tempb{#3}%
1102 \MT@ifempty\@tempb\relax{%
1103 \MT@get@space@unit4%

```

```

1104 \MT@scale@to@em
1105 \shbscode\MT@font\MT@char=\@tempcntb
1106 <debug>\MT@info@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char}%
1107 }%
1108 \MT@ifdefined@c\MT@sp@inh@name{%
1109 \MT@ifdefined@n\MT@inh@\MT@sp@inh@name @\MT@char @}{%
1110 \expandafter\MT@map@tlist@c
1111 \csname MT@inh@\MT@sp@inh@name @\MT@char @\endcsname
1112 \MT@set@sp@heirs
1113 }\relax
1114 }\relax
1115 }

\MT@set@sp@heirs
1116 \def\MT@set@sp@heirs#1{%
1117 \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1118 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1119 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1120 <*debug>
1121 \MT@info@n1{2}{-- heir of \MT@char: #1}%
1122 \MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1123 \number\stbscode\MT@font\MT@char/%
1124 \number\shbscode\MT@font\MT@char}%
1125 </debug>
1126 }

\MT@set@all@sp
\MT@reset@sp@codes 1127 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1128 <debug>\MT@info@n1{3}{-- knbs/stbs/shbs: setting all to \number#1/\number#2/\number#3}%
1129 \@tempcnta=\z@
1130 \MT@while@num{\@tempcnta < \@ccclvi}{%
1131 \knbscode\MT@font\@tempcnta=#1\relax
1132 \stbscode\MT@font\@tempcnta=#2\relax
1133 \shbscode\MT@font\@tempcnta=#3\relax
1134 \advance\@tempcnta \@ne
1135 }%
1136 }
1137 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1138 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1139 \def\MT@preset@sp{%
1140 \expandafter\expandafter\expandafter\MT@preset@sp@
1141 \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1142 }
1143 \def\MT@preset@sp@#1,#2,#3\@nil{%
1144 \ifx\MT@sp@unit@\@empty
1145 \MT@warning@n1{%
1146 Cannot preset characters relative to their widths\MessageBreak
1147 for spacing list '\MT@sp@c@name'. Presetting them\MessageBreak
1148 relative to lem instead}%
1149 \MT@preset@aux@factor{#1}\@tempa
1150 \MT@preset@aux@factor{#2}\@tempc
1151 \MT@preset@aux@factor{#3}\@tempb
1152 \else
1153 \MT@preset@aux@space{#1}\@tempa
1154 \def\@tempb{#2}%
1155 \MT@get@space@unit\thr@@
1156 \MT@scale@to@em
1157 \edef\@tempc{\number\@tempcntb}%

```

```

1158 \def\@tempb{#3}%
1159 \MT@get@space@unit4%
1160 \MT@scale@to@em
1161 \edef\@tempb{\number\@tempcntb}%
1162 \fi
1163 \MT@set@all@sp\@tempa\@tempc\@tempb
1164 }

```

13.4.4 Additional Kerning

`\MT@kerning` Again, only check for additional kerning for new versions of pdfTeX.

```

1165 \ifnum\MT@pdfTeX@no > 5
1166 \def\MT@kerning{\MT@maybe@do{kn}}
1167 \else
1168 \let\MT@kerning\relax
1169 \fi

```

`\MT@set@kn@codes`

```

1170 \def\MT@set@kn@codes{%
1171 \MT@reset@kn@codes
1172 \MT@if@list@exists{%
1173 \MT@get@dimen@six
1174 \MT@get@opt
1175 \MT@get@inh@list
1176 \MT@load@list{\MT@kn@c@name}%
1177 \edef\MT@curr@list@name{kerning list '\MT@kn@c@name'}%
1178 \MT@let@cn\@tempc{\MT@kn@c@\MT@kn@c@name}%
1179 \expandafter\MT@kn@do\@tempc,\relax,%
1180 }\relax
1181 }

```

`\MT@kn@do`

```

1182 \def\MT@kn@do#1,{%
1183 \ifx\relax#1\@empty \else
1184 \MT@kn@split #1=\relax
1185 \expandafter\MT@kn@do
1186 \fi
1187 }

```

`\MT@kn@split`

```

1188 \def\MT@kn@split#1=#2=#3\relax{%
1189 \def\@tempa{#1}%
1190 \ifx\@tempa\@empty \else
1191 \MT@get@slot
1192 \ifnum\MT@char > \m@ne
1193 \MT@get@char@unit
1194 \MT@kn@split@val#2\relax
1195 \fi
1196 \fi
1197 }

```

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

```

1198 \def\MT@kn@split@val#1,#2\relax{%
1199 \def\@tempb{#1}%
1200 \MT@if@empty\@tempb\relax{%
1201 \MT@get@space@unit\tw@
1202 \MT@scale@to@em

```

```

1203 \knbccode\MT@font\MT@char=\@tempcntb
1204 <debug>\MT@info{n1}{4}{;;; knbc (\MT@char): \number\knbccode\MT@font\MT@char}%
1205 }%
1206 \def\@tempb{#2}%
1207 \MT@ifempty\@tempb\relax{%
1208 \MT@get@space@unit\tw@
1209 \MT@scale@to@em
1210 \knaccode\MT@font\MT@char=\@tempcntb
1211 <debug>\MT@info{n1}{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char}%
1212 }%
1213 \MT@ifdefined@c\MT@kn@inh@name{%
1214 \MT@ifdefined{n\MT@inh@\MT@kn@inh@name @\MT@char @}{%
1215 \expandafter\MT@map@tlist@c
1216 \csname MT@inh@\MT@kn@inh@name @\MT@char @\endcsname
1217 \MT@set@kn@heirs
1218 } \relax
1219 } \relax
1220 }

\MT@set@kn@heirs

1221 \def\MT@set@kn@heirs#1{%
1222 \knbccode\MT@font#1=\knbccode\MT@font\MT@char
1223 \knaccode\MT@font#1=\knaccode\MT@font\MT@char
1224 <*debug>
1225 \MT@info{n1}{2}{-- heir of \MT@char: #1}%
1226 \MT@info{n1}{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char/%
1227 \number\knaccode\MT@font\MT@char}%
1228 </debug>
1229 }

\MT@set@all@kn
\MT@reset@kn@codes
\MT@reset@kn@codes@
1230 \def\MT@set@all@kn#1#2{%
1231 <debug>\MT@info{n1}{3}{-- knac/knbc: setting all to \number#1/\number#2}%
1232 \@tempcnta=\z@
1233 \MT@while@num{\@tempcnta < \@ccclvi}{%
1234 \knbccode\MT@font\@tempcnta=#1\relax
1235 \knaccode\MT@font\@tempcnta=#2\relax
1236 \advance\@tempcnta \@ne
1237 }%
1238 }
1239 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1240 \let\MT@reset@kn@codes\relax

\MT@preset@kn
\MT@preset@kn@
1241 \def\MT@preset@kn{%
1242 \expandafter\expandafter\expandafter\MT@preset@kn@
1243 \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1244 }
1245 \def\MT@preset@kn@#1,#2\@nil{%
1246 \ifx\MT@kn@unit@\@empty
1247 \MT@warning@n1{%
1248 Cannot preset characters relative to their widths\MessageBreak
1249 for kerning list '\MT@kn@c@name'. Presetting them\MessageBreak
1250 relative to 1em instead}%
1251 \let\MT@preset@aux\MT@preset@aux@factor
1252 \else
1253 \let\MT@preset@aux\MT@preset@aux@space
1254 \fi

```

Letterspacing factor.

```

1255 \MT@ifstreq\MT@kn@context{letterspacing}{%
1256 \@tempcnta\MT@kn@factor\relax
1257 \MT@scale\@tempcnta \MT@letterspacing@ \@m
1258 \edef\MT@kn@factor@{\number\@tempcnta}%
1259 }\relax
1260 \MT@preset@aux{#1}\@tempa
1261 \MT@preset@aux{#2}\@tempb
1262 \MT@set@all@kn\@tempa\@tempb
1263 }

```

13.4.5 Letterspacing

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

```

1264 \ifnum\MT@pdfTeX@no > 5
1265 \DeclareRobustCommand\lsstyle{%
1266 \ifMT@kerning

```

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

```

1267 \begingroup
1268 \escapechar\m@ne
1269 \expandafter\MT@exp@two@n\expandafter\MT@in@clist
1270 \csname\curr@fontshape/\f@size\expandafter\endcsname
1271 \csname MT@knlist@font@\MT@kn@setname\endcsname
1272 \ifMT@inlist@ \else
1273 \expandafter\MT@xadd
1274 \csname MT@knlist@font@\MT@kn@setname\endcsname
1275 {\csname\curr@fontshape/\f@size\endcsname,}%
1276 \fi
1277 \endgroup
1278 \fi
1279 \MT@kerningtrue
1280 \pdfappendkern\@ne
1281 \pdfprependkern\@ne
1282 \def\MT@kn@setname{all}%
1283 \MT@ifdefinedc\MT@letterspacing@\relax{%
1284 \let\MT@letterspacing@\MT@letterspacing
1285 }%
1286 \microtypecontext{kerning=letterspacing}%
1287 }
1288 \else
1289 \DeclareRobustCommand\lsstyle{%
1290 \MT@warning{Letterspacing only works with pdfTeX version 1.3x\MessageBreak
1291 or newer. You might want to use the 'soul' package\MessageBreak
1292 instead}%
1293 \global\let\lsstyle\relax
1294 }
1295 \fi

```

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

```

1296 \DeclareRobustCommand\textls[2][2]{%
1297 \MT@ifempty{#1}{%
1298 \let\MT@letterspacing@\undefined

```

```

1299 }{%
1300 \KV@sp@def\MT@letterspacing@{#1 }%
1301 }%
1302 {\lsstyle #2}%
1303 }
1304 /beta

```

13.4.6 Disabling Ligatures

`\MT@noligatures` The possibility to disable ligatures is a new features of pdf_T_EX 1.30.0.

```

1305 \ifnum\MT@pdfTeXno < 5 \MT@hop@else@fi{%
1306 \let\MT@noligatures\relax
1307 }\else\MT@hop@fi{%
1308 \def\MT@noligatures{%
1309 \csname ifMT@\MT@abbr@n\endcsname
1310 \MT@dotrue
1311 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1312 \MT@ifdefined@n{\MT@checklist@##1}%
1313 {\csname MT@checklist@##1\endcsname}%
1314 {\MT@checklist@{##1}}%
1315 {n}}%
1316 }%
1317 \else
1318 \MT@dofalse
1319 \fi
1320 \ifMT@do
1321 \pdfnoligatures\MT@font
1322 \MT@vinfo{... Disabling ligatures}%
1323 \fi
1324 }
1325 }\fi

```

13.4.7 Loading the Configuration

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

```

1326 \def\MT@load@list#1{%
1327 \edef\@tempa{#1}%
1328 \MT@let@cn\@tempb{\MT@\MT@feat @c@\@tempa load}%
1329 \MT@ifstreq\@tempa\@tempb{%
1330 \MT@warning{\@nameuse{\MT@abbr@\MT@feat} list '\@tempa' cannot load itself}%
1331 }{%
1332 \ifx\@tempb\relax \else
1333 \MT@ifdefined@n{\MT@\MT@feat @c@\@tempb}{%
1334 \MT@vinfo{... : First loading \@nameuse{\MT@abbr@\MT@feat} list '\@tempb'}%
1335 \begingroup
1336 \MT@load@list{\@tempb}%
1337 \endgroup
1338 \edef\MT@curr@list@name{\@nameuse{\MT@abbr@\MT@feat} list '\@tempb'}%
1339 \MT@let@cn\@tempc{\MT@\MT@feat @c@\@tempb}%
1340 \expandafter\csname MT@\MT@feat @do\expandafter\endcsname\@tempc,\relax,%
1341 }{%
1342 \MT@warning{\@nameuse{\MT@abbr@\MT@feat} list '\@tempb' undefined.
1343 Cannot load\MessageBreak it from list '\@tempa'}%
1344 }%
1345 \fi
1346 }%
1347 }

```

`\MT@find@file` Micro-typographic settings may be written into a file `mt-(font family).cfg`.
`\MT@file@list` We must also record whether we've already loaded the file.

```
1348 \let\MT@file@list\@empty
1349 \def\MT@find@file#1{%
```

Check for existence of the file only once.

```
1350 \MT@in@clist{#1}\MT@file@list
1351 \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.

```
1352 \MT@begin@catcodes
1353 \let\MT@begin@catcodes\relax
1354 \let\MT@end@catcodes\relax
1355 \InputIfFileExists{mt-#1.cfg}{%
1356 \MT@vinfo{... Loading configuration file mt-#1.cfg}%
1357 \MT@xadd\MT@file@list{#1,}%
1358 }{%
1359 \expandafter\MT@get@basefamily#1\relax\relax\relax
1360 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1361 \ifMT@inlist\else
1362 \InputIfFileExists{mt-\@tempa.cfg}{%
1363 \MT@vinfo{... Loading configuration file mt-\@tempa.cfg}%
1364 \MT@xadd\MT@file@list{\@tempa,#1,}%
1365 }{%
1366 \MT@vinfo{... No configuration file mt-#1.cfg}%
1367 \MT@xadd\MT@file@list{#1,}%
1368 }%
1369 \fi
1370 }%
1371 \endgroup
1372 \fi
1373 }
```

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

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

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

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

```
1374 \def\MT@begin@catcodes{%
1375 \begingroup
1376 \makeatletter
1377 \catcode'\^7%
1378 \catcode'\_9%
1379 \catcode'\^^I9%
1380 \catcode'\^^M9%
1381 \catcode'\\z0
1382 \catcode'\{\@ne
1383 \catcode'\}\tw@
1384 \catcode'\#6%
1385 \catcode'\%14%
```

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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```

1386 \MT@map@tlist@n
1387 {!\!"$%&'(\)*+,\-\.\/\:\;\<=\>?[\]\_\'|\|~}%
1388 \@makeother

```

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

```

1389 \def\MT@remove@spaces##1{%
1390 \let\KV@@sp@def\def
1391 }

```

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

```

1392 \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 ...).

```

1393 \def\MT@get@basefamily#1#2#3#4\relax{%
1394 \ifx#2\relax \def\@tempa{#1}\else
1395 \ifx#3\relax \def\@tempa{#1#2}\else
1396 \def\@tempa{#1#2#3}%
1397 \ifx\relax#4\relax \else
1398 \MT@ifstreq{#4}{\string x}\relax{%
1399 \MT@ifstreq{#4}{\string j}\relax{%
1400 \MT@ifstreq{#4}{\string w}\relax{%
1401 \def\@tempa{#1#2#3#4}}}\fi\fi\fi
1402 }

```

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

```

\MT@get@listname@
1403 \def\MT@get@listname#1{%
1404 (debug) \MT@info{n1}{trying to find \@nameuse{MT@abbr@#1} list for font \MT@font}%
1405 \let\MT@listname\undefined
1406 \def\@tempb{#1}%
1407 \MT@map@tlist@c\MT@try@order\MT@get@listname@
1408 }
1409 \def\MT@get@listname@#1{%
1410 \expandafter\MT@next@listname#1%
1411 \ifx\MT@listname\undefined \else
1412 \expandafter\MT@tlist@break
1413 \fi
1414 }

```

\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.

```

1415 \def\MT@try@order{%
1416 {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
1417 {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%

```


1418 }

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

```
1419 \def\MT@next@listname#1#2#3#4{%
1420   \edef\@tempa{\MT@encoding
1421     /\ifnum#1=\@ne \MT@family\fi
1422     /\ifnum#2=\@ne \MT@series\fi
1423     /\ifnum#3=\@ne \MT@shape\fi
1424     /\ifnum#4=\@ne *\fi
1425     \MT@context}%
1426 <debug>\MT@info@n1{1}{trying \@tempa}%
1427 \MT@ifdefined@n{\MT@\@tempb @\@tempa}{%
1428   \MT@next@listname@#4%
1429 }{%
```

Also try with an alias family.

```
1430   \ifnum#1=\@ne
1431     \ifx\MT@familyalias\@empty \else
1432       \edef\@tempa{\MT@encoding
1433         /\MT@familyalias
1434         /\ifnum#2=\@ne \MT@series\fi
1435         /\ifnum#3=\@ne \MT@shape\fi
1436         /\ifnum#4=\@ne *\fi
1437         \MT@context}%
1438 <debug>\MT@info@n1{1}{(alias) \@tempa}%
1439   \MT@ifdefined@n{\MT@\@tempb @\@tempa}{%
1440     \MT@next@listname@#4%
1441   }\relax
1442 \fi
1443 \fi
1444 }%
1445 }
```

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

```
1446 \def\MT@next@listname@#1{%
1447   \ifnum#1=\@ne
1448     \expandafter\MT@in@rlist\csname MT@\@tempb @\@tempa @sizes\endcsname
1449     \ifMT@inlist@
1450       \let\MT@listname\MT@size@name
1451     \fi
1452   \else
1453     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
1454   \fi
1455 }
```

\MT@if@list@exists

```
\MT@context 1456 \def\MT@if@list@exists{%
1457   \expandafter\let\expandafter\MT@context\csname MT@\MT@feat @context\endcsname
1458   \MT@get@listname{\MT@feat @c}%
1459   \MT@ifdefined@c\MT@listname{%
1460     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
1461     \ifMT@nonselected
1462       \MT@vinfo{... Applying non-selected expansion (list '\MT@ex@c@name')}%
1463     \else
1464       \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list
1465         '\@nameuse{MT@\MT@feat @c@name}}}%
1466   \fi
1467   \@firstoftwo
```

```
1468 }{%
```

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

```
1469 \MT@let@nc{MT@MT@feat @c@name}\@empty
```

Don't warn if selected=false.

```
1470 \ifMT@nonselected
1471 \MT@vinfo{... Applying non-selected expansion}%
1472 \else
1473 \MT@warning{I cannot find a \@nameuse{MT@abbr@MT@feat} list
1474 for font\MessageBreak'\MT@exp@string\MT@font'%
1475 \ifx\MT@context\@empty\else\space(context: '\MT@context')\fi.
1476 Switching off\MessageBreak\@nameuse{MT@abbr@MT@feat} for this font}%
1477 \fi
1478 \@secondoftwo
1479 }%
1480 }
```

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

```
\MT@context 1481 \def\MT@get@inh@list{%
1482 \let\MT@context\@empty
1483 \MT@get@listname{\MT@feat @inh}%
1484 \MT@ifdefined@{c}\MT@listname{%
1485 \MT@edef@{n}{MT@MT@feat @inh@name}{\MT@listname}%
1486 <debug>
1487 \MT@dinfo@n{1}{... Using \@nameuse{MT@abbr@MT@feat} inheritance list
1488 '\@nameuse{MT@MT@feat @inh@name}'}%
1489 </debug>
1490 \MT@let@cn@tempc{MT@MT@feat @inh@c\c\name MT@MT@feat @inh@name\endcsname}%
```

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

```
1491 \ifx\@tempc\@empty \else
1492 <debug>\MT@dinfo@n{1}{parsing inheritance list ...}%
1493 \MT@let@cn\MT@inh@name{MT@MT@feat @inh@name}%
1494 \def\MT@curr@list@name{inheritance list}%
1495 \expandafter\MT@inh@do\@tempc,\relax,%
1496 \global\MT@let@nc{MT@MT@feat @inh@c\c\name MT@MT@feat @inh@name\endcsname}\@empty
1497 \fi
1498 }{%
1499 \MT@let@nc{MT@MT@feat @inh@name}\@undefined
1500 }%
1501 }
```

13.4.8 Translating Characters

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

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

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

```
1502 \def\MT@get@slot{%
1503 \escapechar'\
1504 \let\Mt@char\m@ne
1505 \MT@noresttrue
```

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

```
1506 \mt@toks=\expandafter\expandafter\expandafter{\expandafter\string\@tempa}%
1507 \edef\MT@char{\expandafter\meaning\@tempa}%
```

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

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

```
1508 \expandafter\MT@is@letter\@tempa\relax\relax
1509 \ifnum\MT@char < \z@
```

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

```
1510 \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.

```
1511 \MT@ifdefined@n{\MT@encoding\MT@detokenize\@tempa}%
1512 \MT@is@symbol
1513 {%
```

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

```
1514 \expandafter\MT@is@composite\@tempa\relax\relax
1515 }%
1516 \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.

```
1517 \MT@exp@two@c\MT@is@char\MT@char\MT@charstring\relax\relax\relax
1518 \fi
1519 \fi
```

```
1520 \let\MT@char\MT@char
1521 \ifnum\MT@char < \z@
1522 \MT@warn@unknown
1523 \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.

```
1524 \ifMT@norest \else
1525 \MT@warn@unknown@i
1526 \let\MT@char\m@ne
1527 \fi
1528 \fi
1529 \escapechar\m@ne
1530 }
```

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

```
\MT@testrest 1531 \newif\ifMT@norest
1532 \def\MT@testrest#1#2{%
1533 \MT@ifstreq{#1}{#2}\relax\MT@norestfalse
1534 }
```

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

```

1535 \def\MT@is@letter#1#2\relax{%
1536   \ifcat a\noexpand#1\relax
1537     \edef\Mt@char{\number'#1}%
1538     \ifx\\#2\\%
1539 <debug>\MT@info@n1{3}{> '\the\mt@toks' is a letter (\Mt@char)}%
1540     \else
1541       \MT@norestfalse
1542     \fi
1543   \else
1544     \ifcat 1\noexpand#1\relax
1545       \edef\Mt@char{\number'#1}%
1546 <debug>\MT@info@n1{3}{> '\the\mt@toks' is a character (\Mt@char)}%
1547       \ifx\\#2\\%
1548         \ifnum\Mt@char>127 \Mt@warn@ascii \fi
1549       \else
1550         \MT@norestfalse
1551         \expandafter\MT@is@number#1#2\relax\relax
1552       \fi
1553     \fi
1554   \fi
1555 }
```

`\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.

```

1556 \def\MT@is@number#1#2#3\relax{%
1557   \ifx\relax#3\relax \else
1558     \ifx\relax#2\relax \else
1559       \MT@noresttrue
1560       \if#1"\relax
1561         \def\x{\uppercase{\edef\Mt@char{\number#1#2#3}}}\x
1562 <debug>\MT@info@n1{3}{> ... a hexadecimal number: \Mt@char}%
1563       \else
1564         \if#1'\relax
1565           \def\Mt@char{\number#1#2#3}%
1566 <debug> \MT@info@n1{3}{> ... an octal number: \Mt@char}%
1567         \else
1568           \MT@ifnumber{#1#2#3}{%
1569             \def\Mt@char{\number#1#2#3}%
1570 <debug> \MT@info@n1{3}{> ... a decimal number: \Mt@char}%
1571           }\MT@norestfalse
1572         \fi
1573       \fi
1574       \ifnum\Mt@char > \ecclv
1575         \Mt@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
1576         \let\Mt@char\m@ne
1577       \fi
1578     \fi
1579   \fi
1580 }
```

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

Unfortunately, the `inputenc` definitions prefer the protected/generic variants

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

Unicode characters (inputenc/utf8) are currently not supported.

```
1581 \def\MT@is@active#1#2\@nil{%
1582   \ifx\#2\%
1583     \ifnum\catcode'#1 = \active
1584       \begingroup
1585         \set@display@protect
1586         \def\IeC#1{##1}%
```

The character is undefined in the encoding.

```
1587     \def\@inpenc@undefined@##1{undefined^^J%
1588       (microtype)\@spaces\@spaces\@spaces\@spaces
1589       in input encoding ''##1''}%
1590     \edef\x{%
1591       \def\noexpand\@tempa{\@tempa}%
```

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

```
1592     \mt@toks={\the\mt@toks\space (= \@tempa)}}%
1593   }%
1594   \expandafter\endgroup\x
1595   \fi
1596   \fi
1597 }
```

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

```
1598 \def\MT@is@symbol{%
1599   \edef\@tempa{\expandafter
1600     \csname\expandafter
1601       \MT@encoding\expandafter
1602       \string\@tempa
1603     \endcsname}%
1604   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
1605     \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
1606   \ifnum\MT@char < \z@
```

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

```
1607   \expandafter\MT@is@letter\@tempa\relax\relax
1608   \fi
1609 }
```

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

```
\MT@charstring 1610 \debug\def\Mt@info{\MT@info@nl{3}}
1611 \begingroup
1612   \catcode'\/=0
1613   /MT@map@tliston{/C/H/A/R}/@makeother
1614   /lowercase{%
1615     /def/x{%
1616       /def\MT@charstring{\CHAR"%
```

```

1617 /def\Mt@is@char##1\CHAR"##2##3##4/relax{%
1618 /ifx/relax##1/relax
1619 /if##3\relax
1620 /edef\Mt@char{/number"##2}%
1621 /MT@testrest/Mt@charstring{##3##4}%
1622 /else
1623 /edef\Mt@char{/number"##2##3}%
1624 /MT@testrest/Mt@charstring{##4}%
1625 /fi
1626 (debug) /Mt@info{> '/the/mt@toks' is a \char (/Mt@char)}%
1627 /fi
1628 }%
1629 }%
1630 }
1631 /expandafter/endgroup/x

```

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

```

1632 \def\Mt@is@composite#1#2\relax{%
1633 \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.

```

1634 \edef\@tempa{\expandafter
1635 \csname\expandafter
1636 \string\csname\Mt@encoding\endcsname
1637 \string#1-%
1638 \string#2%
1639 \endcsname}%
1640 \expandafter\Mt@is@letter\@tempa\relax\relax
1641 \fi
1642 }

```

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

```

1643 \ifcase\Mt@etex@no
1644 \def\Mt@detokenize#1{\MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty}
1645 \else
1646 \def\Mt@detokenize#1{\detokenize
1647 \expandafter\expandafter\expandafter{\expandafter\string#1}}
1648 \fi

```

Some warning messages, for performance reasons separated here.

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

```

1649 \let\Mt@curr@list@name\@empty

```

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

```

1650 \def\Mt@warn@ascii{%
1651 \MT@warning@nl{Character '\the\mt@toks' (= \Mt@char)
1652 is outside of ASCII range.\MessageBreak
1653 You must load the 'inputenc' package before using\MessageBreak
1654 8-bit characters in \MT@curr@list@name}%
1655 }

```

```

\MT@warn@number@too@large Number too large.
1656 \def\MT@warn@number@too@large#1{%
1657   \MT@warning@nl{%
1658     Number #1 in encoding '\MT@encoding' too large!\MessageBreak
1659     Ignoring it in \MT@curr@list@name}%
1660 }

\MT@warn@unknown@i Not all of the string has been parsed.
1661 \def\MT@warn@unknown@i{%
1662   \MT@warning@nl{%
1663     Unknown slot number of character '\the\mt@toks' in\MessageBreak
1664     font encoding '\MT@encoding'. Make sure it's a single\MessageBreak
1665     character (or a number) in \MT@curr@list@name}%
1666 }

\MT@warn@unknown No idea what went wrong.
1667 \def\MT@warn@unknown{%
1668   \MT@warning@nl{%
1669     Unknown slot number of character '\the\mt@toks' in\MessageBreak
1670     font encoding '\MT@encoding' in \MT@curr@list@name}%
1671 }

```

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

For my reference:

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

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

and were loading fonts, e.g. jurabib, ledmac, pifont (loaded by hyperref), tipa, and probably many more. Furthermore, we had to include a hack for the IEEEtran class which loads all fonts in the class file itself (to fine tune inter-word spacing). Then I learned that even my favorite class, the memoir class, loads fonts. To cut this short: It seemed to get out of hand, and I decided that it would be better to use `\pickup@font` and decide for ourselves whether we've already seen that font. I hope the overhead isn't too large.

`\MT@font` 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.

```
1672 \let\MT@font\empty
1673 \g@addto@macro\do@subst@correction{%
1674   \xdef\MT@font{\csname \curr@fontshape/\f@size\endcsname}%
1675 }
```

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

```
1676 \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
1677 \ifx\pickup@font\MT@orig@pickupfont \else
1678   \MT@warning@n1{%
1679     Command \string\pickup@font\space is not defined as expected.\MessageBreak
1680     Double-check whether micro-typography is indeed\MessageBreak
1681     applied to the document.\MessageBreak (Hint: Turn on 'verbose' mode)%
1682   }
1683 \fi
```

Then we append our stuff.

```
1684 \g@addto@macro\pickup@font{%
1685   \begingroup
1686   \escapechar\m@ne
```

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

```
1687   \ifx\MT@font\empty
1688     \let\MT@font\font@name
1689   \else
1690     \ifx\MT@font\font@name \else
1691       \expandafter\MT@xadd
1692       \csname MT@MT@curr@contexts font@list\endcsname{\font@name,}%
1693     \fi
1694   \fi
```

Comma-separated lists are used to remember the fonts we've already set up. There is one list for each combination of contexts.

```
1695   \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
1696   \csname MT@MT@curr@contexts font@list\endcsname
1697   \ifMT@inlist@ \else
1698     \MT@setupfont
```

Add the font to the current list and remove it from the other contexts' lists.

```
1699     \expandafter\MT@xadd\csname MT@MT@curr@contexts font@list\endcsname{\MT@font,}%
1700     \MT@map@tlist@c\MT@doc@contexts\MT@rem@from@lists
1701   \fi
```



```

1702 \endgroup
1703 \global\let\MT@font\empty
1704 }

\MT@rem@from@lists Recurse through all context font lists of the document and remove the font, unless
it's the current context.
1705 \def\MT@rem@from@lists#1{%
1706 \MT@ifstreq{#1}\MT@curr@contexts\relax{%
1707 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@list
1708 \expandafter\MT@font\csname MT@#1font@list\endcsname
1709 }%
1710 }

\MT@pickupfont Remember the patched command for later.
1711 \let\MT@pickupfont\pickup@font

\MT@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.
1712 \let\MT@add@accent\add@accent
1713 \def\add@accent#1#2{%
1714 \let\pickup@font\MT@orig@pickupfont
1715 \MT@add@accent{#1}{#2}%
1716 \let\pickup@font\MT@pickupfont
1717 }

We also test whether our definition has survived (\pickup@font is at least redef-
ined by CJK16).
1718 \AtBeginDocument{%
1719 \ifx\MT@pickupfont\pickup@font \else
1720 \MT@error{%
1721 Another package has overwritten the definition\MessageBreak
1722 of \string\pickup@font. I might not be able to\MessageBreak
1723 apply any micro-typography. Please find the\MessageBreak
1724 culprit, and load it before the microtype package
1725 }{%
1726 The microtype package attaches the micro-typographic setup to\MessageBreak
1727 \string\pickup@font. If the other package has simply overwritten this\MessageBreak
1728 command, nothing will work. If, on the other hand, it has changed\MessageBreak
1729 the command in a cautious way, everything may be fine.\MessageBreak
1730 In either case, please send a report to <w.m.l@gmx.net>.
1731 }%
1732 \fi
1733 }

```

16 Therefore, microtype will probably not work together with CJK. However, I would be glad to be proven wrong.

13.5 Configuration

13.5.1 Font Sets

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

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

```

1734 \def\DeclareMicrotypeSet{%
1735   \ifstar
1736     {\ifnextchar[\MT@DeclareSetAndUseIt
1737       {\MT@DeclareSetAndUseIt[]}}%
1738     {\ifnextchar[\MT@DeclareSet
1739       {\MT@DeclareSet[]}}%
1740 }

\MT@DeclareSet
\MT@DeclareSetAndUseIt 1741 \def\MT@DeclareSet[#1]{%
1742   \MT@DeclareSet@{#1}%
1743 }
1744 \def\MT@DeclareSetAndUseIt[#1]#2#3{%
1745   \MT@DeclareSet@{#1}{#2}{#3}%
1746   \UseMicrotypeSet[#1]{#2}%
1747 }

\MT@DeclareSet@
1748 \def\MT@DeclareSet@#1#2#3{%
1749   \KV@sp@def\@tempa{#1}%
1750   \MT@ifempty\@tempa{%
1751     \MT@declare@sets{pr}{#2}{#3}%
1752     \MT@declare@sets{ex}{#2}{#3}%
1753     \beta
1754     \MT@declare@sets{sp}{#2}{#3}%
1755     \MT@declare@sets{kn}{#2}{#3}%
1756     /beta
1757   }{%
1758     \MT@map@cclist@c\@tempa{%
1759       {\KV@sp@def\@tempa{##1}%
1760        \MT@ifempty\@tempa{\relax}%
1761        \MT@exp@one@n\MT@declare@sets
1762         {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}}}%
1763     }%
1764   }%
1765 }

```

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

```
1766 \let\MT@curr@set@name\empty
```

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

```

1767 \def\MT@declare@sets#1#2#3{%
1768   \KV@sp@def\MT@curr@set@name{#2}%
1769   \MT@ifdefinedn{MT@#1@set@@\MT@curr@set@name}{%
1770     \MT@warning{Redefining set '\MT@curr@set@name'}%
1771   }\relax
1772   \global\MT@let@nc{MT@#1@set@@\MT@curr@set@name}\empty

```

```

1773 <debug>\MT@info{1}{declaring \@nameuse{MT@abbr@#1} set '\MT@curr@set@name'}%
1774 \setkeys{MT@#1@set}{#3}%
1775 }

```

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

```

1776 \def\MT@define@set@keys#1{%
1777 \MT@define@set@key@{encoding}{#1}%
1778 \MT@define@set@key@{family}{#1}%
1779 \MT@define@set@key@{series}{#1}%
1780 \MT@define@set@key@{shape}{#1}%
1781 \MT@define@set@key@size{#1}%
1782 \MT@define@set@key@font{#1}%
1783 }

```

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

```

1784 \def\MT@define@set@key@#1#2{%
1785 \csname MT@#2list@#1@MT@curr@set@name\endcsname
1786 \define@key{MT@#2@set}{#1}[]{}%
1787 \global\MT@let@nc{MT@#2list@#1@MT@curr@set@name}\@empty
1788 \MT@map@clist@n{##1}{%

```

Get rid of spaces.

```

1789 \KV@@sp@def\MT@val{####1}%
1790 \MT@get@highlevel{#1}%
1791 \MT@make@string\MT@val
1792 \expandafter\MT@xadd
1793 \csname MT@#2list@#1@MT@curr@set@name\endcsname{\MT@val,%}
1794 }%
1795 <debug>\MT@info@n{1}{-- #1: \@nameuse{MT@#2list@#1@MT@curr@set@name}}%
1796 }%
1797 }

```

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

```

1798 \def\MT@get@highlevel#1{%
1799 \expandafter\MT@test@ast\MT@val*\@nil{%

```

And ‘family = *’ will become \familydefault.

```

1800 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
1801 \edef\MT@val{\csname \@tempa default\endcsname}%

```

Beware that \rmdefault etc. might change after this package has been loaded! Therefore, we check (once for each characteristic/list) at the beginning of the document.

```

1802 \ifx\nodocument\relax \else
1803 \expandafter\ifx
1804 \csname MT@check@MT@curr@set@name @\@tempa\endcsname\@empty
1805 \else
1806 \global\MT@edef@n{MT@MT@curr@set@name @\@tempa @default}{\MT@val}%
1807 \edef\x{{\MT@curr@set@name}{\@tempa}}%
1808 \MT@exp@one@n\AtBeginDocument{%
1809 \expandafter\MT@check@default\x
1810 }%
1811 \global\MT@let@nc{MT@check@MT@curr@set@name @\@tempa}\@empty
1812 \fi
1813 \fi
1814 }%
1815 }

```

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

```
1816 \def\MT@test@ast#1*#2\@nil{%
1817   \def\@tempa{#1}%
1818   \MT@ifempty{#2}%
1819     \@gobble
1820     \@firstofone
1821 }
```

\MT@check@default

```
1822 \def\MT@check@default#1#2{%
1823   \MT@let@cn\@tempa{MT@#1@#2@default}%
1824   \edef\@tempb{\csname #2default\endcsname}%
1825   \ifx\@tempa\@tempb \else
1826     \MT@warning@n{%
1827       \expandafter\noexpand\csname #2default\endcsname
1828       has changed ('@tempa' <> '@tempb')!\MessageBreak
1829       This might affect the '#1' font set.\MessageBreak
1830       Please make all relevant font changes *before*\MessageBreak
1831       loading the 'microtype' package}%
1832   \fi
1833 }
```

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

```
1834 \def\MT@define@set@key@size#1{%
1835   \define@key{MT@#1@set}{size}[]{%
1836     \MT@map@clist@n{##1}{%
1837       \KV@spdef\MT@val{###1}%
1838       \expandafter\MT@get@range\MT@val--\@nil
1839       \ifx\MT@val\relax \else
1840         \expandafter\MT@xadd
1841         \csname MT@#1list@size@\MT@curr@set@name\endcsname
1842         {{{\MT@lower}{\MT@upper}\relax}}%
1843       \fi
1844     }%
1845   <debug>\MT@dinfo@n{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
1846   }%
1847 }
```

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

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

```
\MT@lower 1848 \def\MT@get@range#1-#2-#3\@nil{%
1849   \MT@ifempty{#1}{%
1850     \MT@ifempty{#2}{%
1851       \let\MT@val\relax
1852     }%
1853     \def\MT@lower{0}%
1854     \def\MT@val{#2}%
1855     \MT@get@size
1856     \edef\MT@upper{\MT@val}%
1857   }%
1858   }%
1859   \def\MT@val{#1}%
```

```

1860 \MT@get@size
1861 \ifx\MT@val\relax \else
1862 \edef\MT@lower{\MT@val}%
1863 \MT@ifempty{#2}{%
1864 \MT@ifempty{#3}{%
1865 \def\MT@upper{-1}%
1866 }{%
1867 \def\MT@upper{2048}%
1868 }%
1869 }{%
1870 \def\MT@val{#2}%
1871 \MT@get@size
1872 \ifx\MT@val\relax \else
1873 \MT@ifgt\MT@lower\MT@val{%
1874 \MT@warning{%
1875 Invalid size range (\MT@lower\space > \MT@val) in font set
1876 '\MT@curr@set@name'.\MessageBreak Swapping sizes}%
1877 \edef\MT@upper{\MT@lower}%
1878 \edef\MT@lower{\MT@val}%
1879 }{%
1880 \edef\MT@upper{\MT@val}%
1881 }%
1882 \MT@ifeq\MT@lower\MT@upper{%
1883 \def\MT@upper{-1}%
1884 }\relax
1885 \fi
1886 }%
1887 \fi
1888 }%
1889 }

```

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

```
1890 \def\MT@get@size{%
```

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

```

1891 \if*\MT@val\relax
1892 \def\@tempa{\normalsize}%
1893 \else
1894 \MT@let@cn\@tempa{\MT@val}%
1895 \fi
1896 \ifx\@tempa\relax \else

```

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

```

1897 \begingroup
1898 \def\set@fontsize##1##2##3##4\@nil{\gdef\MT@val{##2}}%
1899 \@tempa\@nil
1900 \endgroup
1901 \fi

```

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

```

1902 \MT@ifdimen\MT@val{%
1903 \@defaultunits\@tempdima\MT@val pt\relax\@nnil
1904 \edef\MT@val{\strip@pt\@tempdima}%

```

```

1905 }{%
1906 \MT@warning{Could not parse font size '\MT@val'\MessageBreak
1907           in font set '\MT@curr@set@name'}%
1908 \let\MT@val\relax
1909 }%
1910 }

\MT@define@set@key@font

1911 \def\MT@define@set@key@font#1{%
1912 \define@key{MT@#1@set}{font}[]{%
1913 \MT@map@clist@n{##1}{%
1914 \KV@sp@def\MT@val{###1}%
1915 \expandafter\MT@get@font\MT@val///// \@nil
1916 \expandafter\MT@xadd
1917 \csname MT@#1list@font@\MT@curr@set@name\endcsname
1918 {\csname\MT@val\endcsname,}%
1919 }%
1920 (debug)\MT@info@n1{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
1921 }%
1922 }

```

\MT@get@font Translate any asterisks.

```

1923 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
1924 \MT@ifempty{#1#2#3#4#5}\relax{%
1925 \let\@tempb\@empty
1926 \def\MT@temp{#1/#2/#3/#4/#5}%
1927 \MT@get@axis{encoding}{#1}%
1928 \MT@get@axis{family}{#2}%
1929 \MT@get@axis{series}{#3}%
1930 \MT@get@axis{shape}{#4}%
1931 \MT@ifempty{#5}{%
1932 \MT@warning{size axis is empty in font specification\MessageBreak
1933 '\MT@temp'. Using \string\normalsize\space instead}%
1934 \def\MT@val{*}%
1935 }{%
1936 \def\MT@val{#5}%
1937 }%
1938 \MT@get@size
1939 \ifx\MT@val\relax\def\MT@val{0}\fi
1940 \edef\MT@val{\expandafter\@gobble\@tempb/\MT@val}%
1941 }%
1942 }

```

\MT@get@axis

```

1943 \def\MT@get@axis#1#2{%
1944 \def\MT@val{#2}%
1945 \MT@get@highlevel{#1}%
1946 \MT@ifempty\MT@val{%
1947 \MT@warning{#1 axis is empty in font specification\MessageBreak
1948 '\MT@temp'. Using '\csname #1default\endcsname' instead}%
1949 \edef\@tempb{\@tempb/\csname #1default\endcsname}%
1950 }{%
1951 \edef\@tempb{\@tempb/\MT@val}%
1952 }%
1953 }

```

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

```

1954 \MT@define@set@keys{pr}
1955 \MT@define@set@keys{ex}

```

```

1956 <beta>
1957 \MT@define@set@keys{sp}
1958 \MT@define@set@keys{kn}
1959 </beta>

```

It is also used for \DisableLigatures.

```
1960 \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.

```

1961 \renewcommand*\UseMicrotypeSet[2] [] {%
1962   \KV@@sp@def\@tempa{#1}%
1963   \MT@ifempty\@tempa{%
1964     \MT@use@set{pr}{#2}%
1965     \MT@use@set{ex}{#2}%
1966 <beta>
1967     \MT@use@set{sp}{#2}%
1968     \MT@use@set{kn}{#2}%
1969 </beta>
1970   }{%
1971     \MT@map@clist@c\@tempa{%
1972       {\KV@@sp@def\@tempa{##1}%
1973        \MT@ifempty\@tempa\relax}%
1974       \MT@exp@one@n\MT@use@set{\csname MT@rbba@\@tempa\endcsname}{#2}}}%
1975   }%
1976 }%
1977 }

```

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

```

\MT@ex@setname 1978 \def\MT@use@set#1#2{%
\MT@use@set 1979   \KV@@sp@def\@tempa{#2}%
1980   \MT@ifdefinedn{MT@#1@set@@\@tempa}{%
1981     \global\MT@edefn{MT@#1@setname}{\@tempa}%
1982     \MT@info{Using \@nameuse{MT@abbr@#1} set '\@tempa'}%
1983   }{%
1984     \MT@ifdefinedn{MT@#1@setname}\relax{%
1985       \global\MT@edefn{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
1986     }%
1987     \MT@warning{%
1988       The \@nameuse{MT@abbr@#1} set '\@tempa' is undeclared.\MessageBreak
1989       Using set '\@nameuse{MT@#1@setname}' instead}%
1990   }%
1991 }

```

\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.

```

1992 \renewcommand*\DeclareMicrotypeSetDefault[2] [] {%
1993   \KV@@sp@def\@tempa{#1}%
1994   \MT@ifempty\@tempa{%
1995     \MT@set@default@set{pr}{#2}%
1996     \MT@set@default@set{ex}{#2}%
1997 <beta>
1998     \MT@set@default@set{sp}{#2}%
1999     \MT@set@default@set{kn}{#2}%
2000 </beta>
2001   }{%
2002     \MT@map@clist@c\@tempa{%
2003       {\KV@@sp@def\@tempa{##1}%
2004        \MT@ifempty\@tempa\relax}%

```

```

2005      \MT@exp@one@n\MT@set@default@set
2006      {\csname MT@rbba@{\@tempa\endcsname}{#2}}}%
2007    }%
2008  }%
2009 }

\MT@default@pr@set
\MT@default@ex@set 2010 \def\MT@set@default@set#1#2{%
\MT@default@kn@set 2011 \KV@sp@def\@tempa{#2}%
\MT@default@sp@set 2012 \MT@ifdefined@n{MT@#1@set@{\@tempa}}{%
\MT@set@default@set 2013 debug\MT@info{1}{declaring default \@nameuse{MT@abbr@#1} set '\@tempa'}%
2014 \global\MT@edef@n{MT@default@#1@set}{\@tempa}%
2015 }{%
2016 \MT@warning{%
2017   The \@nameuse{MT@abbr@#1} set '\@tempa' is not declared.\MessageBreak
2018   Cannot make it the default set. Using set\MessageBreak 'all' instead}%
2019 \global\MT@edef@n{MT@default@#1@set}{all}%
2020 }%
2021 }

```

`\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.

```

2022 \renewcommand*\DeclareMicrotypeAlias[2]{%
2023   \KV@sp@def\@tempa{#1}%
2024   \KV@sp@def\@tempb{#2}%
2025   \MT@make@string\@tempb
2026   \MT@ifdefined@n{MT@\@tempa @alias}{%
2027     \MT@warning{Alias font family '\@tempb' will override
2028       alias '\@nameuse{MT@\@tempa @alias}'\MessageBreak
2029       for font family '\@tempa'}}\relax
2030   \global\MT@edef@n{MT@\@tempa @alias}{\@tempb}%

```

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

```

2031 \MT@ifdefined@c\MT@family{%
2032 debug\MT@info{1}{Activating alias font '\@tempb' for '\MT@family'}%
2033 \global\let\MT@familyalias\@tempb
2034 }\relax
2035 }

```

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

```

2036 \def\LoadMicrotypeFile#1{%
2037   \KV@sp@def\@tempa{#1}%
2038   \MT@make@string\@tempa
2039   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2040   \ifMT@inlist@
2041     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2042   \else
2043     \MT@xadd\MT@file@list{\@tempa,}%
2044     \MT@begin@catcodes
2045     \InputIfFileExists{mt-\@tempa.cfg}{%
2046       \MT@vinfo{... Loading configuration file mt-\@tempa.cfg}%
2047     }{%
2048       \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2049         does not exist}%
2050     }%
2051     \MT@end@catcodes
2052   \fi

```


2053 }

`\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'.

```

2054 \ifnum\MT@pdfTeX@no > 4
2055   \renewcommand*\DisableLigatures[1]{%
2056     \MT@noligaturestrue
2057     \MT@declare@sets{nl}{no ligatures}{#1}%
2058     \gdef\MT@nl@setname{no ligatures}%
2059   }
2060 \else
  If pdfTeX is too old, we issue a warning and neutralize the command.
2061   \renewcommand*\DisableLigatures[1]{%
2062     \MT@warning{Disabling ligatures of a font is only possible\MessageBreak
2063       with pdfTeX version 1.30 or later.\MessageBreak
2064       Ignoring \string\DisableLigatures}%
2065     \let\DisableLigatures\@gobble
2066   }
2067 \fi

```

13.5.2 Interaction with babel

`\DeclareMicrotypeBabelHook` Declare the context that should be loaded when a babel language is selected.

```

2068 <*\beta>
2069 \def\DeclareMicrotypeBabelHook#1#2{%
2070   \MT@map@clist@n{#1}{%
2071     \KV@sp@def\@tempa{##1}%
2072     \global\MT@def@n{MT@babel@\@tempa}{#2}%
2073   }%
2074 }
2075 \onlypreamble{\DeclareMicrotypeBabelHook}
2076 </\beta>
2077 </package>

```

13.5.3 Declarations

We now set up some default sets (in the main configuration file).

```

2078 <*\config&m-t>
2079
2080 %%% -----
2081 %%% FONT SETS
2082
2083 \DeclareMicrotypeSet{all}
2084   { }
2085
2086 \DeclareMicrotypeSet{allmath}
2087   { encoding = {T1,LY1,OT1,OT4,T5,TS1,OML,OMS,U} }
2088
2089 \DeclareMicrotypeSet{alltext}
2090   { encoding = {T1,LY1,OT1,OT4,T5,TS1} }
2091
2092 \DeclareMicrotypeSet{basicmath}
2093   { encoding = {T1,LY1,OT1,OT4,T5,OML,OMS},
2094     family   = {rm*,sf*},
2095     series    = {m},

```

```

2096     size      = {normalsize,footnotesize,small,large}
2097   }
2098
2099 \DeclareMicrotypeSet{basictext}
2100   { encoding = {T1,LY1,OT1,OT4,T5},
2101     family   = {rm*,sf*},
2102     series   = {m},
2103     size     = {normalsize,footnotesize,small,large}
2104   }
2105
2106 \DeclareMicrotypeSet{normalfont}
2107   { font = */*/*/*/* }
2108

```

The default sets.

```

2109 %%% -----
2110 %%% DEFAULT SETS
2111
2112 \DeclareMicrotypeSetDefault[protrusion]{alltext}
2113 \DeclareMicrotypeSetDefault[expansion]{basictext}
2114 < *beta >
2115 \DeclareMicrotypeSetDefault[spacing]{basictext}
2116 \DeclareMicrotypeSetDefault[kerning]{alltext}
2117 < /beta >
2118

```

The Latin Modern fonts, the virtual fonts from the ae and zefonts, and the eco and hfoldsty packages (oldstyle numerals) all inherit the (basic) settings from Computer Modern Roman. Some of them are in part overwritten later. The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively.

```

2119 %%% -----
2120 %%% FONT ALIASES
2121
2122 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
2123 \DeclareMicrotypeAlias{aer}{cmr} % ae
2124 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
2125 \DeclareMicrotypeAlias{cmor}{cmr} % eco
2126 \DeclareMicrotypeAlias{hfor}{cmr} % hfoldsty
2127 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
2128 \DeclareMicrotypeAlias{qpl}{ppl} % qfonts/QuasiPalatino
2129 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
2130 \DeclareMicrotypeAlias{qtm}{ptm} % qfonts/QuasiTimes
2131

```

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

```

2132 < *beta >

```

Interaction with babel.

```

2133 %%% -----
2134 %%% INTERACTION WITH THE 'babel' PACKAGE
2135
2136 \DeclareMicrotypeBabelHook
2137   {french,francais}
2138   {kerning=french, spacing=}
2139

```

```

2140 \DeclareMicrotypeBabelHook
2141   {english,american,USenglish,british,UKenglish}
2142   {kerning=, spacing=nonfrench}
2143
2144 \DeclareMicrotypeBabelHook
2145   {turkish}
2146   {kerning=turkish, spacing=}
2147
2148 </beta>
2149 </config&m-t>

```

13.5.4 Fine Tuning

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

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

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

`\MT@load`

`\MT@extra@factor`

`\MT@extra@preset` 2150 *<package>*

`\MT@extra@unit` 2151 `\renewcommand*\SetProtrusion[2][]{%`

`\MT@extra@context` 2152 `\let\MT@pr@c@name\undefined`

2153 `\let\MT@load\undefined`

2154 `\let\MT@extra@factor\undefined`

2155 `\let\MT@extra@unit\undefined`

2156 `\let\MT@extra@preset\undefined`

2157 `\let\MT@extra@context\empty`

Parse the optional first argument:

```
2158 \setkeys{MT@pr@c}{#1}%
```

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

```

2159 \MT@get@codes@name{pr}%
2160 \MT@set@pr@opt
2161 <debug>\MT@dinfo{1}{creating protrusion list '\MT@pr@c@name'}%
2162 \def\MT@permutelist{pr}%
2163 \setkeys{MT@pr@c}{#2}%

```

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

```
2164 \MT@permute
```

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

```

2165 \MT@begin@catcodes
2166 \MT@set@pr@list
2167 }

```

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

```

2168 \def\MT@set@pr@list#1{%
2169   \global\MT@def@n{MT@pr@c@MT@pr@c@name}{#1}%
2170   \MT@end@catcodes
2171 }

```

```

\SetExpansion \SetExpansion only differs in that it allows some extra options (stretch, shrink,
\MT@ex@name step, auto).
\MT@load 2172 \renewcommand*\SetExpansion[2] [] {%
2173 \let\MT@ex@name\@undefined
\MT@extra@factor 2174 \let\MT@load\@undefined
2175 \let\MT@extra@factor\@undefined
\MT@extra@stretch 2176 \let\MT@extra@stretch\@undefined
2177 \let\MT@extra@shrink\@undefined
\MT@extra@step 2178 \let\MT@extra@step\@undefined
2179 \let\MT@extra@auto\@undefined
\MT@extra@preset 2180 \let\MT@extra@preset\@undefined
2181 \let\MT@extra@context\@empty
\MT@extra@context 2182 \setkeys{MT@ex@c}{#1}%
2183 \MT@get@codes@name{ex}%
2184 \MT@set@ex@opt
2185 <debug>\MT@info{1}{creating expansion list '\MT@ex@name'}%
2186 \def\MT@permutelist{ex@c}%
2187 \setkeys{MT@ex@c}{#2}%
2188 \MT@permute
2189 \MT@begin@catcodes
2190 \MT@set@ex@list
2191 }

\MT@set@ex@list Same story.
2192 \def\MT@set@ex@list#1{%
2193 \global\MT@def@n{MT@ex@c@MT@ex@name}{#1}%
2194 \MT@end@catcodes
2195 }
2196 <beta>

\SetExtraSpacing
\MT@sp@c@name 2197 \renewcommand*\SetExtraSpacing[2] [] {%
2198 \let\MT@sp@c@name\@undefined
\MT@load 2199 \let\MT@load\@undefined
\MT@extra@factor 2200 \let\MT@extra@factor\@undefined
\MT@extra@unit 2201 \let\MT@extra@unit\@undefined
\MT@extra@preset 2202 \let\MT@extra@preset\@undefined
2203 \let\MT@extra@context\@empty
\MT@extra@context 2204 \setkeys{MT@sp@c}{#1}%
2205 \MT@get@codes@name{sp}%
2206 \MT@set@sp@opt
2207 <debug>\MT@info{1}{creating space list '\MT@sp@c@name'}%
2208 \def\MT@permutelist{sp@c}%
2209 \setkeys{MT@sp@c}{#2}%
2210 \MT@permute
2211 \MT@begin@catcodes
2212 \MT@set@sp@list
2213 }

\MT@set@sp@list
2214 \def\MT@set@sp@list#1{%
2215 \global\MT@def@n{MT@sp@c@MT@sp@c@name}{#1}%
2216 \MT@end@catcodes
2217 }

\SetExtraKerning
\MT@kn@c@name 2218 \renewcommand*\SetExtraKerning[2] [] {%
\MT@load 2219 \let\MT@kn@c@name\@undefined
\MT@extra@factor
\MT@extra@unit
\MT@extra@preset
\MT@extra@context

```

```

2220 \let\MT@load\@undefined
2221 \let\MT@extra@factor\@undefined
2222 \let\MT@extra@unit\@undefined
2223 \let\MT@extra@preset\@undefined
2224 \let\MT@extra@context\@empty
2225 \setkeys{MT@kn@c}{#1}%
2226 \MT@get@codes@name{kn}%
2227 \MT@set@kn@opt
2228 <debug>\MT@dinfo{1}{creating kerning list '\MT@kn@c@name'}%
2229 \def\MT@permutelist{kn}%
2230 \setkeys{MT@kn@c}{#2}%
2231 \MT@permute
2232 \MT@begin@catcodes
2233 \MT@set@kn@list
2234 }

```

\MT@set@kn@list

```

2235 \def\MT@set@kn@list#1{%
2236   \global\MT@defn{MT@kn@c@\MT@kn@c@name}{#1}%
2237   \MT@end@catcodes
2238 }
2239 </beta>

```

\MT@get@codes@name Simply use a roman number as the list name if the user didn't bother creating one.

```

2240 \def\MT@get@codes@name#1{%
2241   \MT@ifdefinedn{MT@#1@c@name}{%
2242     \MT@ifdefinedn{MT@#1@c@\csname MT@#1@c@name\endsname}{%
2243       \MT@warning{Redefining list '\@nameuse{MT@#1@c@name}'}%
2244     }%
2245   }{%
2246     \@tempcnta=\@ne
2247     \MT@while@num{\@tempcnta > \z@}{%
2248       \MT@ifdefinedn{MT@#1@c@#1-\romannumeral\@tempcnta}{%
2249         \advance \@tempcnta \@ne
2250       }{%
2251         \MT@edefn{MT@#1@c@name}{#1-\romannumeral\@tempcnta}%
2252         \@tempcnta=\z@
2253       }%
2254     }%
2255   }%
2256   \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%

```

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

```

2257 \MT@ifdefinedc\MT@load{%
2258   \global\MT@let@nc{MT@#1@c@\MT@curr@set@name load}\MT@load
2259 }%
2260 }

```

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

```

2261 \def\MT@set@pr@opt{%
2262   \MT@set@opt@{pr}{factor}%
2263   \MT@set@opt@{pr}{unit}%
2264   \MT@set@opt@{pr}{preset}%
2265 }

```

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

```

2266 \def\MT@set@ex@opt{%
2267   \MT@ifdefinedc\MT@extra@factor{%

```

```

2268 \ifnum\MT@extra@factor>\@m
2269 \MT@warning@n\{Expansion factor \number\MT@extra@factor\space too
2270 large in list\MessageBreak '\MT@ex@cc@name'. Setting it to the
2271 maximum of 1000}%
2272 \let\MT@extra@factor\@m
2273 \fi
2274 \global\MT@let@nc{\MT@ex@cc@\MT@ex@cc@name @factor}\MT@extra@factor
2275 }\relax
2276 \MT@set@opt@{ex}{stretch}%
2277 \MT@set@opt@{ex}{shrink}%
2278 \MT@set@opt@{ex}{step}%
2279 \MT@set@opt@{ex}{auto}%
2280 \MT@set@opt@{ex}{preset}%
2281 }
2282 (*beta)

\MT@set@sp@opt
2283 \def\MT@set@sp@opt{%
2284 \MT@set@opt@{sp}{factor}%
2285 \MT@set@opt@{sp}{unit}%
2286 \MT@set@opt@{sp}{preset}%
2287 }

\MT@set@kn@opt
2288 \def\MT@set@kn@opt{%
2289 \MT@set@opt@{kn}{factor}%
2290 \MT@set@opt@{kn}{unit}%
2291 \MT@set@opt@{kn}{preset}%
2292 }
2293 (/beta)

\MT@set@opt@
2294 \def\MT@set@opt@#1#2{%
2295 \MT@ifdefinedn{\MT@extra@#2}{%
2296 \global\MT@let@nn{\MT@#1@c@\csname MT@#1@c@name\endcsname @#2}{\MT@extra@#2}%
2297 }\relax
2298 }

\MT@define@code@key Define the keys for expansion and protrusion character code lists.
2299 \def\MT@define@code@key#1#2{%
2300 \define@key{MT@#2}{#1}[]{%
2301 \@tempcnta=\@ne
2302 \MT@map@clist@n{##1}{%
2303 \KV@sp@def\MT@val{####1}%

Here, too, we allow for something like ‘bf*’. It will be expanded immediately.
2304 \MT@get@highlevel{#1}%
2305 \MT@edef@n{\MT@temp#1\romannumeral\@tempcnta}{\MT@val}%
2306 \advance\@tempcnta \@ne
2307 }%
2308 }%
2309 }

\MT@define@code@key@size
2310 \def\MT@define@code@key@size#1{%
2311 \define@key{MT@#1}{size}[]{%
2312 \MT@map@clist@n{##1}{%
2313 \KV@sp@def\MT@val{####1}%

```

```

2314 \expandafter\MT@get@range\MT@val--\@nil
2315 \ifx\MT@val\relax \else
2316 \expandafter\MT@xadd
2317 \csname MT@tempsize\endcsname
2318 {{{\MT@lower}{\MT@upper}{\csname MT@#1@name\endcsname}}}%
2319 \fi
2320 }%
2321 }%
2322 }

```

\MT@define@code@key@font

```

2323 \def\MT@define@code@key@font#1{%
2324 \define@key{MT@#1}{font}[]{}%
2325 \MT@map@clist@n{##1}{%
2326 \KV@sp@def\MT@val{###1}%
2327 \expandafter\MT@get@font@and@size\MT@val////\@nil
2328 \global\MT@edef@n{MT@MT@permutelist @\@tempb}%
2329 {\csname MT@MT@permutelist @name\endcsname}%
2330 \expandafter\MT@xaddb
2331 \csname MT@MT@permutelist @\@tempb @size\endcsname
2332 {{{\MT@val}{\m@ne}{\csname MT@#1@name\endcsname}}}%
2333 }%
2334 }%
2335 }

```

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

```

2336 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2337 \MT@ifempty{#1#2#3#4#5}\relax{%
2338 \let\@tempb\empty
2339 \def\MT@temp{#1/#2/#3/#4/#5}%
2340 \MT@get@axis{encoding}{#1}%
2341 \MT@get@axis{family}{#2}%
2342 \MT@get@axis{series}{#3}%
2343 \MT@get@axis{shape}{#4}%

```

Remove leading slash and append an asterisk for the size.

```

2344 \edef\@tempb{\expandafter\@gobble\@tempb/*}%
2345 \MT@ifempty{#5}{%
2346 \MT@warning{size axis is empty in font specification\MessageBreak
2347 '\MT@temp'. Using \string\normalsize\space instead}%
2348 \def\MT@val{*}%
2349 }{%
2350 \def\MT@val{#5}%
2351 }%
2352 \MT@get@size
2353 }%
2354 }

```

\MT@declare@codes

```

2355 \def\MT@declare@codes#1{%
2356 \define@key{MT@#1@c}{name}[]{}%
2357 \MT@ifempty{##1}\relax{%
2358 \MT@def@n{MT@#1@c@name}{##1}%
2359 }%
2360 }%
2361 \define@key{MT@#1@c}{load}[]{}%
2362 \MT@ifempty{##1}\relax{%
2363 \def\MT@load{##1}%
2364 }%

```

```

2365 }%
2366 \define@key{MT@#1c}{factor}[]{%
2367   \MT@ifempty{##1}\relax{%
2368     \def\MT@extra@factor{##1}%
2369   }%
2370 }%
2371 \MT@define@code@key{encoding}{#1c}%
2372 \MT@define@code@key{family}{#1c}%
2373 \MT@define@code@key{series}{#1c}%
2374 \MT@define@code@key{shape}{#1c}%
2375 \MT@define@code@key{size}{#1c}%
2376 \MT@define@code@key{font}{#1c}%
2377 \define@key{MT@#1c}{preset}[]{%
2378   \MT@ifempty{##1}\relax{%
2379     \def\MT@extra@preset{##1}%
2380   }%
2381 }%

```

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

```

2382 \define@key{MT@#1c}{context}[]{%
2383   \MT@ifempty{##1}\relax{%
2384     \def\MT@extra@context{##1}%
2385   }%
2386 }%
2387 }
2388 \MT@declare@codes{pr}
2389 \MT@declare@codes{ex}
2390 <beta>
2391 \MT@declare@codes{sp}
2392 \MT@declare@codes{kn}
2393 /beta>

```

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

```

2394 \define@key{MT@pr@c}{unit}[character]{%
2395   \let\MT@extra@unit\empty
2396   \KV@sp@def\@tempa{#1}%
2397   \MT@ifstreq\@tempa{relative}{%
2398     \MT@warning{Value 'relative' for key 'unit' is deprecated.\MessageBreak
2399       Use 'unit=character' instead. For now, I'll do it.\MessageBreak
2400       for you}%
2401     \def\@tempa{character}%
2402   }\relax
2403   \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.

```

2404   \MT@ifdimen\@tempa{%
2405     \let\MT@extra@unit\@tempa
2406   }{%
2407     \MT@warning{'\@tempa' is not a dimension.\MessageBreak
2408       Ignoring it and setting values relative to\MessageBreak
2409       character widths}%
2410   }%
2411 }%
2412 }

```

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

```

2413 <beta>
2414 \def\MT@define@key@unit#1{%

```



```

2415 \define@key{MT@#1@c}{unit}[space]{%
2416 \let\MT@extra@unit\@empty
2417 \KV@sp@def\@tempa{##1}%
2418 \MT@ifstreq\@tempa{relative}{%
2419 \MT@warning{Value 'relative' for key 'unit' is deprecated.\MessageBreak
2420 Use 'unit=character' instead. For now, I'll do it.\MessageBreak
2421 for you}%
2422 \def\@tempa{character}%
2423 }\relax
2424 \MT@ifstreq\@tempa{character}\relax{%
2425 \let\MT@extra@unit\m@ne
2426 \MT@ifstreq\@tempa{space}\relax{%
2427 \MT@ifdimen\@tempa{%
2428 \let\MT@extra@unit\@tempa
2429 }{%
2430 \MT@warning{'\@tempa' is not a dimension.\MessageBreak
2431 Ignoring it and setting values relative to\MessageBreak
2432 width of space}%
2433 }%
2434 }%
2435 }%
2436 }%
2437 }

2438 \MT@define@key@unit{sp}
2439 \MT@define@key@unit{kn}
2440 /beta

```

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

```

2441 \def\MT@define@ex@c@key#1{%
2442 \define@key{MT@ex@c}{#1}[]{%
2443 \MT@ifempty{##1}\relax{%
2444 \MT@ifnumber{##1}{%

```

A space terminates the number.

```

2445 \MT@defon{MT@extra@#1}{##1 }%
2446 }{%
2447 \MT@warning{%
2448 Value '##1' for option '#1' is not a number.\MessageBreak
2449 Ignoring it}%
2450 }%
2451 }%
2452 }%
2453 }

```

```

2454 \MT@define@ex@c@key{stretch}
2455 \MT@define@ex@c@key{shrink}
2456 \MT@define@ex@c@key{step}
2457 \define@key{MT@ex@c}{auto}[true]{%
2458 \KV@sp@def\@tempa{##1}%
2459 \csname if\@tempa\endcsname

```

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

```

2460 \ifnum\MT@pdftex@no > \thr@@
2461 \def\MT@extra@auto{autoexpand}%
2462 \else
2463 \MT@warning{pdfTeX too old for automatic font expansion}%
2464 \fi
2465 \else
2466 \ifnum\MT@pdftex@no > \thr@@
2467 \let\MT@extra@auto\@empty

```

```

2468 \fi
2469 \fi
2470 }

```

13.5.5 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.

```

2471 \renewcommand*\DeclareCharacterInheritance[1] [] {%
2472 \KV@sp@def\@tempa{#1}%
2473 \MT@begin@catcodes
2474 \MT@set@inh@list
2475 }

```

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

```

2476 \def\MT@set@inh@list#1#2{%
2477 \MT@ifempty\@tempa{%
2478 \MT@declare@char@inh{pr}{#1}{#2}%
2479 \MT@declare@char@inh{ex}{#1}{#2}%
2480 <beta>
2481 \MT@declare@char@inh{sp}{#1}{#2}%
2482 \MT@declare@char@inh{kn}{#1}{#2}%
2483 </beta>
2484 }{%
2485 \MT@map@clist@c\@tempa{%
2486 {\KV@sp@def\@tempa{##1}%
2487 \MT@ifempty\@tempa\relax{%
2488 \MT@exp@one@n\MT@declare@char@inh
2489 {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}}}%
2490 }%
2491 }%
2492 \MT@end@catcodes
2493 }

```

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

```

2494 \def\MT@declare@char@inh#1#2#3{%
2495 \MT@let@nc{MT@#1@inh@name}\@undefined
2496 \MT@get@inh@name{#1}%
2497 <debug> \MT@dinfo{1}{creating inheritance list '\@nameuse{MT@#1@inh@name}'}%
2498 \global\MT@def@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
2499 \def\MT@permutelist{#1@inh}%
2500 \setkeys{MT@#1@inh}{#2}%
2501 \MT@permute
2502 }

```

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

```

2503 \def\MT@get@inh@name#1{%
2504 \@tempcnta=\@ne
2505 \MT@while@num{\@tempcnta > \z@}{%
2506 \MT@ifdefined@n{MT@#1@inh@#1-inh-\romannumeral\@tempcnta}{%
2507 \advance \@tempcnta \@ne

```

```

2508 }{%
2509     \MT@edef\MT@#1@inh@name}{#1-inh-\romannumeral\@tempcnta}%
2510     \@tempcnta=\z@
2511 }%
2512 }%
2513 }

```

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

```

2514 \def\MT@define@inh@key@encoding#1{%
2515     \define@key{MT@#1}{encoding}[]{%
2516         \def\MT@val{#1}%
2517         \expandafter\MT@encoding@check\MT@val,\@nil
2518         \MT@get@highlevel{encoding}%
2519         \MT@edef\MT@tempencoding\romannumeral1{\MT@val}%
2520     }%
2521 }

```

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

```

2522 \def\MT@encoding@check#1,#2\@nil{%
2523     \MT@ifempty{#2}\relax{%
2524         \edef\MT@val{#1}%
2525         \MT@warning{You may only specify one encoding for character\MessageBreak
2526                     inheritance lists. Ignoring encoding(s) #2}%
2527     }%
2528 }

```

`\MT@define@inh@keys`

```

2529 \def\MT@define@inh@keys#1{%
2530     \MT@define@inh@key@encoding{#1@inh}%

```

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

```

2531     \MT@define@code@key{family}{#1@inh}%
2532     \MT@define@code@key{series}{#1@inh}%
2533     \MT@define@code@key{shape}{#1@inh}%
2534     \MT@define@code@key@size{#1@inh}%
2535     \MT@define@code@key@font{#1@inh}%
2536 }
2537 \MT@define@inh@keys{pr}
2538 \MT@define@inh@keys{ex}
2539 *beta
2540 \MT@define@inh@keys{sp}
2541 \MT@define@inh@keys{kn}
2542 /beta

```

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

```

2543 \def\MT@inh@do#1,{%
2544     \ifx\relax#1\@empty \else
2545         \MT@inh@split #1==\relax
2546         \expandafter\MT@inh@do
2547     \fi
2548 }

```

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

```

MT@set@{feature}@codes),
2549 \def\MT@inh@split#1=#2=#3\relax{%
2550   \def\@tempa{#1}%
2551   \ifx\@tempa\@empty \else
2552     \MT@get@slot
2553     \ifnum\MT@char > \m@ne
2554       \let\MT@val\MT@char
2555       \MT@map@clist@n{#2}{%
2556         \def\@tempa{##1}%
2557         \ifx\@tempa\@empty \else
2558           \MT@get@slot
2559           \ifnum\MT@char > \m@ne
2560             \expandafter\MT@xadd
2561             \csname MT@inh@\MT@inh@name @\MT@val @\endcsname
2562             {{\MT@char}}%
2563           \fi
2564         \fi
2565       }%
2566   <debug>
2567   \MT@dinfo@n{2}{children of #1 (\MT@val):
2568     \nameuse{MT@inh@\MT@inh@name @\MT@val @}}%
2569 </debug>
2570   \fi
2571 \fi
2572 }

```

13.5.6 Permutation

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

```

2573 \def\MT@permute{%
2574   \let\MT@cnt@encoding\@ne
2575   \MT@permute@

```

Undefine commands for the next round.

```

2576   \MT@permute@reset
2577 }
2578 \def\MT@permute@{%
2579   \let\MT@cnt@family\@ne
2580   \MT@permute@@
2581   \MT@increment\MT@cnt@encoding
2582   \MT@ifdefined@n{\MT@tempencoding\romannumeral\MT@cnt@encoding}%
2583   \MT@permute@
2584   \relax
2585 }
2586 \def\MT@permute@@{%
2587   \let\MT@cnt@series\@ne
2588   \MT@permute@@@
2589   \MT@increment\MT@cnt@family
2590   \MT@ifdefined@n{\MT@tempfamily\romannumeral\MT@cnt@family}%
2591   \MT@permute@@
2592   \relax
2593 }

```

```

2594 \def\MT@permute@@@{%
2595   \let\MT@cnt@shape\@ne
2596   \MT@permute@@@
2597   \MT@increment\MT@cnt@series
2598   \MT@ifdefined@n{\MT@tempseries\romannumeral\MT@cnt@series}%
2599   \MT@permute@@@
2600   \relax
2601 }
2602 \def\MT@permute@@@{%
2603   \MT@permute@@@
2604   \MT@increment\MT@cnt@shape
2605   \MT@ifdefined@n{\MT@tempshape\romannumeral\MT@cnt@shape}%
2606   \MT@permute@@@
2607   \relax
2608 }
2609 \def\MT@permute@@@{%
2610   \MT@permute@define{encoding}%
2611   \MT@permute@define{family}%
2612   \MT@permute@define{series}%
2613   \MT@permute@define{shape}%
2614   \edef\@tempa{\MT@tempencoding
2615               /\MT@tempfamily
2616               /\MT@tempseries
2617               /\MT@tempshape
2618               /\MT@ifdefined@c\MT@tempsize *\@empty}%

```

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

```

2619 \def\@tempb{////}%
2620 \ifx\@tempa\@tempb \else
2621   \ifx\MT@tempencoding\@empty
2622     \MT@warning{%
2623       You have to specify an encoding for\MessageBreak
2624       \@nameuse{MT@abbr@\MT@permutelist} list
2625       '\@nameuse{MT@\MT@permutelist @name}'.\MessageBreak
2626       Ignoring it}%
2627   \else
2628     \MT@ifdefined@c\MT@tempsize{%

```

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

```

2629   \MT@ifdefined@n{\MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}{%
2630     \MT@map@tlist@c
2631     \MT@tempsize
2632     \MT@check@rlist
2633   }\relax
2634   \expandafter\MT@xaddb
2635   \csname MT@\MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
2636   \MT@tempsize
2637   \<debug>
2638   \MT@dinfo@n1{1}{initializing: use list for font \@tempa,\MessageBreak
2639     sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
2640       @sizes\endcsname}%
2641   \</debug>
2642   }{%

```

Only one list should apply to a given combination.

```

2643   \MT@ifdefined@n{\MT@\MT@permutelist @\@tempa\MT@extra@context}{%
2644     \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
2645       '\@nameuse{MT@\MT@permutelist @name}' will override list\MessageBreak
2646       '\@nameuse{MT@\MT@permutelist @\@tempa\MT@extra@context}' for font '\@tempa'}%

```

```

2647     }\relax
2648 <*debug>
2649     \MT@info@n1{1}{initializing: use list for font \@tempa
2650                     \ifx\MT@extra@context\@empty\else\MessageBreak
2651                     (context: \MT@extra@context)\fi}%
2652 </debug>
2653     }%
2654     \global\MT@edef@n{MT@MT@permutelist @\@tempa\MT@extra@context}%
2655     {\csname MT@MT@permutelist @name\endcsname}%
2656     \fi
2657     \fi
2658 }

```

\MT@permute@define Define the commands.

```

2659 \def\MT@permute@define#1{%
2660   \expandafter\@tempcnta=\csname MT@cnt@#1\endcsname\relax
2661   \MT@ifdefined@n{MT@temp#1\romannumeral\@tempcnta}%
2662   {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\romannumeral\@tempcnta\endcsname}}%
2663   {\MT@let@nc{MT@temp#1}\@empty}%
2664 }

```

\MT@permute@reset Reset the commands.

```

2665 \def\MT@permute@reset{%
2666   \MT@permute@reset@{encoding}%
2667   \MT@permute@reset@{family}%
2668   \MT@permute@reset@{series}%
2669   \MT@permute@reset@{shape}%
2670   \let\MT@tempsize\@undefined
2671 }

```

\MT@permute@reset@

```

2672 \def\MT@permute@reset@#1{%
2673   \@tempcnta=\@ne
2674   \MT@loop
2675   \MT@let@nc{MT@temp#1\romannumeral\@tempcnta}\@undefined
2676   \advance\@tempcnta\@ne
2677   \MT@ifdefined@n{MT@temp#1\romannumeral\@tempcnta}%
2678   \iftrue
2679     \iffalse
2680     \MT@repeat
2681 }

```

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

```

2682 \def\MT@check@rlist#1{%
2683   \expandafter\MT@check@rlist@#1%
2684 }

```

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

```

2685 \def\MT@check@rlist@#1#2#3{%
2686   \def\@tempb{#1}%
2687   \def\@tempc{#2}%
2688   \@tempswafalse
2689   \expandafter\MT@map@tlist@c
2690   \csname MT@MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
2691   \MT@check@range
2692 }

```

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

```

2693 \def\MT@check@range#1{%
2694   \expandafter\MT@check@range@#1%
2695 }

```

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

```

2696 \def\MT@check@range@#1#2#3{%
2697   \MT@ifeq{#2}\m@ne{%
2698     \MT@ifeq\@tempc\m@ne{%

```

- Both items are simple sizes.

```

2699         \MT@ifeq\@tempb{#1}\@tempswatruerelax
2700     }{%

```

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

```

2701         \MT@ifgt\@tempb{#1}\relax{%
2702         \MT@ifgt\@tempc{#1}{%
2703             \@tempswatruerelax
2704             \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
2705         }\relax
2706     }%
2707 }%
2708 }{%
2709     \MT@ifeq\@tempc\m@ne{%

```

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

```

2710         \MT@iflt\@tempb{#2}{%
2711         \MT@iflt\@tempb{#1}\relax\@tempswatruerelax
2712     }\relax
2713 }{%

```

- Both items are ranges.

```

2714         \MT@iflt\@tempb{#2}{%
2715         \MT@ifgt\@tempc{#1}{%
2716             \@tempswatruerelax
2717             \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
2718         }\relax
2719     }\relax
2720 }%
2721 }%
2722 \if@tempswa
2723   \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
2724     '\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
2725     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.

```

2726   \expandafter\MT@tlist@break
2727   \fi
2728 }

```

13.6 User Command

\microtypesetup This command may be used anywhere in the document. It accepts the options: protrusion, expansion and activate, and spacing and kerning. Specifying font

sets is not allowed.

```

2729 \def\microtypesetup{\setkeys{MTX}}
2730 \def\MT@define@optionX#1#2{%
2731   \define@key{MTX}{#1}[true]{%
2732     \KV@esp@def\@tempb{#1}%
2733     \MT@map@clist@n{##1}{%
2734       \KV@esp@def\MT@val{###1}%
2735       \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
2736       \MT@ifempty\MT@val\relax{%
2737         \@tempcnta=\m@ne
2738         \MT@ifstreql\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.

```

2739   \MT@checksetup\@tempb{%
2740     \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
2741     \MT@info{Enabling #1
2742       (level \number\csname MT@\@tempb @level\endcsname)}%
2743   }%
2744 }{%
2745   \MT@ifstreql\MT@val{false}{%
2746     \@tempcnta=\z@
2747     \MT@info{Disabling #1}%
2748   }{%
2749     \MT@ifstreql\MT@val{compatibility}{%
2750       \MT@checksetup\@tempb{%
2751         \@tempcnta=\@ne
2752         \MT@let@nc{MT@\@tempb @level}\@ne
2753         \MT@info{Setting #1 to level 1}%
2754       }%
2755     }{%
2756       \MT@ifstreql\MT@val{nocompatibility}{%
2757         \MT@checksetup\@tempb{%
2758           \@tempcnta=\tw@
2759           \MT@let@nc{MT@\@tempb @level}\tw@
2760           \MT@info{Setting #1 to level 2}%
2761         }%
2762       }{%
2763         \MT@warning{%
2764           Value '\MT@val' for key '#1' not recognized.\MessageBreak
2765           Use any of 'true', 'false', 'compatibility' or\MessageBreak
2766           'nocompatibility'}%
2767         }%
2768       }%
2769     }%
2770   }%
2771   \ifnum\@tempcnta>\m@ne
2772     #2\@tempcnta\relax
2773   \fi
2774 }%
2775 }%
2776 }%
2777 }

```

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

```

2778 \def\MT@checksetup#1{%
2779   \expandafter\csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
2780   \expandafter\@firstofone

```



```

2781 \else
2782 \MT@warning{%
2783   You cannot enable \@nameuse{MT@abbr@#1} if it was disabled\MessageBreak
2784   in the package options,}%
2785 \expandafter\@gobble
2786 \fi
2787 }

2788 \MT@define@optionX{protrusion}\pdfprotrudechars
2789 \MT@define@optionX{expansion}\pdfadjustspacing
2790 /beta

```

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

```

2791 \def\MT@define@optionX@#1#2{%
2792   \define@key{MTX}{#1}[true]{%
2793     \KV@sp@def\@tempb{#1}%
2794     \MT@map@clist@n{##1}{%
2795       \KV@sp@def\MT@val{###1}%
2796       \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
2797       \MT@ifempty\MT@val\relax{%
2798         \@tempcnta=\m@ne
2799         \MT@ifstreq\MT@val{true}{%
2800           \MT@checksetup\@tempb{%
2801             \@tempcnta=\@ne
2802             \MT@info{Enabling #1}%
2803           }%
2804         }%
2805         \MT@ifstreq\MT@val{false}{%
2806           \@tempcnta=\z@
2807           \MT@info{Disabling #1}%
2808         }%
2809         \MT@warning{%
2810           Value '\MT@val' for key '#1' not recognized.\MessageBreak
2811           Use either 'true' or 'false'}%
2812       }%
2813     }%
2814     \ifnum\@tempcnta>\m@ne
2815       #2\relax
2816     \fi
2817   }%
2818 }%
2819 }%
2820 }

2821 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
2822 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
2823   \pdfappendkern \@tempcnta}
2824 /beta
2825 \define@key{MTX}{activate}[]{%
2826   \setkeys{MTX}{protrusion={#1}}%
2827   \setkeys{MTX}{expansion={#1}}%
2828 }

```

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

```

2829 \def\microtypecontext#1{%
2830   \setkeys{MTC}{#1}%
2831   \edef\MT@curr@contexts{\MT@pr@context|
2832     \MT@ex@context|
2833     \MT@sp@context|

```

```

2834 \MT@kn@context}%
2835 <debug>\MT@info{2}{>>> current context: \MT@curr@contexts}%

Keep track of all contexts in the document.
2836 \MT@exp@one@n\MT@in@tlist\MT@curr@contexts\MT@doc@contexts
2837 \ifMT@inlist@ \else
2838 \MT@xadd\MT@doc@contexts{{\MT@curr@contexts}}%
2839 <debug>\MT@info{2}{>>> document contexts: \MT@doc@contexts}%
2840 \fi
2841 \selectfont
2842 \aftergroup\MT@reset@context
2843 }

```

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

```

2844 \def\MT@reset@context{%
2845 \MT@vinfo{Resetting contexts on line \the\inputlineno}%
2846 \selectfont
2847 }

```

\MT@define@context

```

2848 \def\MT@define@context#1{%
2849 \define@key{MTC}{#1}[]{}%
2850 \KV@esp@def\@tempb{#1}%
2851 \edef\@tempb{\@nameuse{MT@rbba@\@tempb}}%
2852 \KV@esp@def\MT@val{##1}%
2853 \MT@vinfo{--- Changing #1 context to '\MT@val'}%
2854 \MT@edef@n{MT@\@tempb @context}{\MT@val}%

```

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

```

2855 \MT@ifempty\MT@val\relax{%
2856 \global\MT@let@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes}%
2857 }%
2858 }%
2859 }

2860 \MT@define@context{protrusion}
2861 \MT@define@context{expansion}
2862 <*beta>
2863 \MT@define@context{spacing}
2864 \MT@define@context{kerning}
2865 </beta>

```

\MT@pr@context Initialize the contexts.

```

\MT@ex@context 2866 \let\MT@pr@context\@empty
\MT@sp@context 2867 \let\MT@ex@context\@empty
                2868 <*beta>
\MT@kn@context 2869 \let\MT@sp@context\@empty
\MT@curr@contexts 2870 \let\MT@kn@context\@empty
\MT@doc@contexts 2871 </beta>
                2872 \def\MT@curr@contexts{|||}
\MT@extra@context 2873 \def\MT@doc@contexts{{|||}}
                2874 \let\MT@extra@context\@empty

```

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

```

2875 \def\MT@gobblethree#1#2#3{}
2876 \let\MT@saved@setupfont\MT@setupfont
2877 \define@key{MTX}{disable}[]{}%
2878 \MT@info{Inactivate microtype package}%

```

```

2879 \let\MT@setupfont\MT@gobblethree
2880 }
2881 \define@key{MTX}{enable}[]{}%
2882 \MT@info{Reactivate microtype package}%
2883 \let\MT@setupfont\MT@saved@setupfont
2884 }

```

13.7 Package Options

13.7.1 Declaring the Options

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

```

\ifMT@opt@auto 2885 \newif\ifMT@opt@expansion
2886 \newif\ifMT@opt@auto

```

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

```

2887 \def\MT@define@option#1{%
2888   \define@key{MT}{#1}[true]{}%
2889   \csname MT@opt@#1true\endcsname
2890   \MT@map@clist@n{##1}{}%
2891   \KV@sp@def\MT@val{###1}%
2892   \MT@ifempty\MT@val\relax{}%
2893   \csname MT@#1true\endcsname
2894   \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2895   \MT@ifstreq\MT@val{true}\relax
2896   {%
2897     \MT@ifstreq\MT@val{false}{}%
2898     \csname MT@#1false\endcsname
2899   }{}%
2900   \MT@ifstreq\MT@val{compatibility}{}%
2901   \MT@let@nc{MT@\@tempb @level}\@ne
2902   }{}%
2903   \MT@ifstreq\MT@val{nocompatibility}{}%
2904   \MT@let@nc{MT@\@tempb @level}\tw@
2905   }{}%

```

If everything failed, it should be a set name.

```

2906   \MT@ifdefinedn{MT@\@tempb @set@@\MT@val}{}%
2907   \global\MT@edefn{MT@\@tempb @setname}{\MT@val}%
2908   }{}%
2909   \global\MT@edefn{MT@\@tempb @setname}%
2910   {\@nameuse{MT@default@\@tempb @set}}{}%
2911   \MT@warning@n1{%
2912     The #1 set '\MT@val' is undeclared.\MessageBreak
2913     Using set '\@nameuse{MT@\@tempb @setname}' instead}%
2914   }{}%
2915   }{}%
2916   }{}%
2917   }{}%
2918   }{}%
2919   }{}%
2920   }{}%
2921   }{}%
2922 }

2923 \MT@define@option{protrusion}
2924 \MT@define@option{expansion}

```

activate is a shortcut for protrusion and expansion (and spacing?).

```

2925 \define@key{MT}{activate}[]{}%
2926 \setkeys{MT}{protrusion={#1}}%
2927 \setkeys{MT}{expansion={#1}}%
2928 }
2929 <beta>

```

\MT@define@option@

```

2930 \def\MT@define@option@#1{%
2931 \define@key{MT}{#1}[true]{}%
2932 \csname MT@opt@#1true\endcsname
2933 \MT@map@clist@n{##1}{%
2934 \KV@sp@def\MT@val{###1}%
2935 \MT@ifempty\MT@val\relax{%
2936 \csname MT@#1true\endcsname
2937 \edef\@tempb{\csname MT@rbba@#1\endcsname}%
2938 \MT@ifstreq\MT@val{true}\relax
2939 {%
2940 \MT@ifstreq\MT@val{false}{%
2941 \csname MT@#1false\endcsname
2942 }%
2943 \MT@ifdefinedn{MT@\@tempb @set@\MT@val}{%
2944 \global\MT@edefn{MT@\@tempb @setname}{\MT@val}%
2945 }%
2946 \global\MT@edefn{MT@\@tempb @setname}%
2947 {\@nameuse{MT@default@\@tempb @set}}%
2948 \MT@warning@n{%
2949 The #1 set '\MT@val' is undeclared.\MessageBreak
2950 Using set '\@nameuse{MT@\@tempb @setname}' instead}%
2951 }%
2952 }%
2953 }%
2954 }%
2955 }%
2956 }%
2957 }

2958 \MT@define@option@{spacing}
2959 \MT@define@option@{kerning}
2960 </beta>

```

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

```

2961 \def\MT@def@bool@opt#1{%
2962 \define@key{MT}{#1}[]{}%
2963 \MT@ifempty{##1}%
2964 {\def\@tempa{true}}%
2965 {\def\@tempa{##1}}%
2966 \MT@ifstreq\@tempa{true}\relax{%
2967 \MT@ifstreq\@tempa{false}\relax{%
2968 \MT@warning@n{%
2969 '##1' is not an admissible value for option\MessageBreak
2970 '#1'. Assuming 'false'}%
2971 \def\@tempa{false}%
2972 }%
2973 }%
2974 \csname MT@#1\@tempa\endcsname
2975 }%
2976 }

```

```

2977 \MT@map@tlist@n{{draft}}{DVIoutput}{auto}{selected}%
2978 \beta{babel}%
2979 }\MT@def@bool@opt

```

`final` is the opposite to `draft`.

```

2980 \define@key{MT}{final}[]{}%
2981 \MT@draftfalse
2982 \MT@ifempty{#1}%
2983   {\def\@tempa{true}}%
2984   {\def\@tempa{#1}}%
2985 \MT@ifstreq\@tempa{true}\relax{%
2986   \MT@ifstreq\@tempa{false}%
2987   \MT@drafttrue
2988   {%
2989     \MT@warning@n1{%
2990       '#1' is not an admissible value for option\MessageBreak
2991       'final'. Assuming 'true'}%
2992     \MT@draftfalse
2993   }%
2994 }%
2995 }

```

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

```

2996 \define@key{MT}{verbose}[]{}%
2997 \let\MT@vinfo\MT@info@n1
2998 \MT@ifempty{#1}%
2999   {\def\@tempa{true}}%
3000   {\def\@tempa{#1}}%
3001 \MT@ifstreq\@tempa{true}\relax{%

```

Take problems seriously.

```

3002   \MT@ifstreq\@tempa{errors}{%
3003     \let\MT@warning\MT@warn@err
3004     \let\MT@warning@n1\MT@warn@err
3005   }{%
3006     \let\MT@vinfo\@gobble
3007     \MT@ifstreq\@tempa{false}\relax{%
3008       \MT@warning@n1{%
3009         '#1' is not an admissible value for option\MessageBreak
3010         'verbose'. Assuming 'false'}%
3011     }%
3012   }%
3013 }%
3014 }

```

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

```

3015 \def\MT@def@num@opt#1{%
3016   \define@key{MT}{#1}[]{}%
3017   \MT@ifempty{#1}%
3018     {\MT@let@cn\@tempa{MT@#1@default}}%
3019     {\def\@tempa{#1}}%

```

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

```

3020   \MT@ifnumber\@tempa{%
3021     \MT@edef@n{MT@#1}{\@tempa}%
3022   }{\MT@warning@n1{%
3023     Value '#1' for option '#1' is not a number.\MessageBreak
3024     Using default value of \number\@nameuse{MT@#1@default}}%
3025   }%
3026 }%

```

```

3027 }
3028 \MT@map@tlist@n{{stretch}{shrink}{step}%
3029 beta}{letterspacing}%
3030 }\MT@def@num@opt

factor will define the protrusion factor only.
3031 \define@key{MT}{factor}[]{}%
3032 \MT@ifempty{#1}%
3033 {\let\@tempa\MT@factor@default}%
3034 {\def\@tempa{#1}}%
3035 \MT@ifnumber\@tempa{%
3036 \MT@edef\MT@pr@factor{\@tempa}%
3037 }\MT@warning@n1{%
3038 Value '#1' for option 'factor' is not a number.\MessageBreak
3039 Using default value of \number\MT@factor@default}%
3040 }%
3041 }

```

Unit for codes.

```

3042 \define@key{MT}{unit}[]{}%
3043 \MT@ifempty{#1}%
3044 {\def\@tempa{character}}%
3045 {\KV@sp@def\@tempa{#1}}%
3046 \MT@ifstreq\@tempa{relative}{%
3047 \MT@warning{Value 'relative' for option 'unit' is deprecated.\MessageBreak
3048 Use 'unit=character' instead. For now, I'll do it.\MessageBreak
3049 for you}%
3050 \def\@tempa{character}%
3051 }\relax
3052 \MT@ifstreq\@tempa{character}\relax{%
3053 \MT@ifdimen\@tempa{%
3054 \let\MT@pr@unit\@tempa
3055 }{%
3056 \MT@warning@n1{'\@tempa' is not a dimension. Ignoring it and.\MessageBreak
3057 setting values relative to character widths}%
3058 }%
3059 }%
3060 }

```

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 .)

```

3061 \MT@protrusiontrue
3062 \ifnum\pdfoutput=\z@ \else

```

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

```

3063 \ifnum\MT@pdftex@no > \thr@@
3064 \MT@expansiontrue
3065 \MT@autottrue
3066 \fi
3067 \fi

```

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

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

```

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

```

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

```

3068 \define@key{MT}{config}[]{\relax}
3069 \def\MT@get@config#1config=#2,#3\@nil{%
3070   \MT@ifempty{#2}%
3071     {\def\MT@config@file{microtype.cfg}}%
3072     {\KV@sp\def\MT@config@file{#2.cfg}}%
3073 }
3074 \expandafter\expandafter\expandafter\MT@get@config
3075   \csname opt\@currname.\@currentx\endcsname,config=\@nil

```

Load the file.

```

3076 \IfFileExists{\MT@config@file}{%
3077   \MT@info@nl{Loading configuration file \MT@config@file}%
3078   \MT@begin@catcodes
3079   \let\MT@begin@catcodes\relax
3080   \let\MT@end@catcodes\relax
3081   \input{\MT@config@file}%
3082   \endgroup
3083 }{%
3084   \MT@warning@nl{%
3085     Could not find configuration file '\MT@config@file'\MessageBreak
3086     This will almost certainly cause undesired results.\MessageBreak
3087     Please fix your installation}%
3088 }

```

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

```

3089 \MT@ifdefined@c\MT@default@pr@set\relax{\gdef\MT@default@pr@set{all}}
3090 \MT@ifdefined@c\MT@default@ex@set\relax{\gdef\MT@default@ex@set{all}}
3091 \beta
3092 \MT@ifdefined@c\MT@default@sp@set\relax{\gdef\MT@default@sp@set{all}}
3093 \MT@ifdefined@c\MT@default@kn@set\relax{\gdef\MT@default@kn@set{all}}
3094 /beta
3095 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
3096   \MT@warning{%
3097     The command \string\DeclareMicrotypeSetDefault\space may only\MessageBreak
3098     be used inside the main configuration file.\MessageBreak
3099     Ignoring it}%
3100 }

```

13.7.2 Hook for Other Packages

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

This hook overcomes 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 is not viable), and (2) checking `\AtBeginDocument` can be too late, since fonts might already have been loaded, and consequently set up, in the preamble.

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

```

\def\MinionPro@MT@Hook
{ \DeclareMicrotypeAlias{MinionPro-LF}{MinionPro} }
\@ifundefined{Microtype@Hook}

```

```
{\let\Microtype@Hook\MinionPro@MT@Hook}
{\gaddto@macro\Microtype@Hook{\MinionPro@MT@Hook}}
```

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

```
3101 \MT@ifdefined@c\MicroType@Hook{%
3102   \MT@warning@nl{%
3103     Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3104     Use \string\Microtype@Hook\space instead}\MicroType@Hook}\relax
3105 \MT@ifdefined@c\Microtype@Hook\Microtype@Hook\relax
```

13.7.3 Processing the Options

\MT@ProcessOptionsWithKV Parse options.

```
3106 \def\MT@ProcessOptionsWithKV#1{%
3107   \let\@tempc\relax
3108   \let\KV@tempa\@empty
3109   \MT@map@clist@c\@classoptionslist{%
3110     \def\CurrentOption{##1}%
3111     \MT@ifdefined@n{KV@#1\CurrentOption}{%
3112       \edef\KV@tempa{\KV@tempa,\CurrentOption,}%
3113       \expandtwoargs\@removeelement\CurrentOption
3114       \@unusedoptionlist\@unusedoptionlist
3115     }\relax
3116   }%
3117   \edef\KV@tempa{%
3118     \noexpand\setkeys{#1}{%
3119       \KV@tempa\@optionlist{\@currname.\@current}%
3120     }%
3121   }%
3122   \KV@tempa
3123   \AtEndOfPackage{\let\@unprocessedoptions\relax}%
3124   \let\CurrentOption\@empty
3125 }

3126 \MT@ProcessOptionsWithKV{MT}
```

Now we can take the appropriate actions:

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

Some packages depend on the value of \pdfoutput and will get confused if it is changed after they have been loaded. These packages are, among others: color, graphics, hyperref, crop, contour, pstricks and, as a matter of course, i fpdf. Instead of testing for each package (that's not our job), we issue a different message if \pdfoutput is actually changed by DVIoutput. That must be sufficient!

```
3127 \ifMT@DVIoutput
3128   \ifnum\pdfoutput=\z@
3129     \MT@info@nl{Generating DVI output}
3130   \else
3131     \pdfoutput\z@
3132     \MT@info@nl{Changing output mode to DVI}
```

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


```

3133 \ifMT@opt@expansion \else
3134 \MT@expansionfalse
3135 \fi
3136 \fi
3137 \else
3138 \MT@info@n1{Generating \ifnum\pdfoutput=\z@ DVI \else PDF \fi output}
3139 \fi

```

Tell the log file which options the user has chosen (in case it's interested). We disable most of what we've just defined in the 3139 lines above if we are running in draft mode.

```

3140 \ifMT@draft
3141 \MT@warning@n1{'draft' option active.\MessageBreak
3142             Disabling all micro-typographic extensions.\MessageBreak
3143             This might lead to different line and page breaks}
3144 \MT@protrusionfalse
3145 \MT@expansionfalse
3146 *beta
3147 \MT@spacingfalse
3148 \MT@kerningfalse
3149 /beta
3150 \let\MT@setupfont\relax
3151 \def\DeclareMicrotypeSet{%
3152     \ifstar
3153     {\@ifnextchar[\MT@DeclareSet{\MT@DeclareSet[]}}%
3154     {\@ifnextchar[\MT@DeclareSet{\MT@DeclareSet[]}}%
3155 }
3156 \def\MT@DeclareSet[#1]#2#3{}
3157 \renewcommand*{\UseMicrotypeSet[2] [] {}
3158 \renewcommand*{\SetProtrusion[3] [] {}
3159 \renewcommand*{\SetExpansion[3] [] {}
3160 *beta
3161 \renewcommand*{\SetExtraSpacing[3] [] {}
3162 \renewcommand*{\SetExtraKerning[3] [] {}
3163 /beta
3164 \renewcommand*{\DeclareCharacterInheritance[3] [] {}
3165 \renewcommand*{\DeclareMicrotypeAlias[2] {}
3166 \renewcommand*{\LoadMicrotypeFile[1] {}
3167 \renewcommand*{\microtypesetup[1] {}
3168 \renewcommand*{\microtypecontext[1] {}
3169 \expandafter
3170 \endinput
3171 \fi
3172 \ifMT@protrusion
3173 \pdfprotrudechars\MT@pr@level
3174 \MT@info@n1{Character protrusion enabled (level \number\MT@pr@level)%
3175 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3176     factor: \number\MT@pr@factor\fi
3177 \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}

```

We have to make sure that font sets are active at the end of the package. If the user didn't activate any in the package options, we use those sets declared by `\DeclareMicrotypeSetDefault`. They can still be overridden later on, of course.

```

3178 \MT@ifdefined@c\MT@pr@setname{%
3179 \MT@info@n1{Using protrusion set '\MT@pr@setname'}%
3180 }{%
3181 \global\let\MT@pr@setname\MT@default@pr@set
3182 \MT@info@n1{Using default protrusion set '\MT@pr@setname'}%
3183 }

```

```

3184 \else
3185   \let\MT@protrusion\relax
3186   \MT@info@n1{No character protrusion}
3187 \fi
3188 \ifMT@expansion

```

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

```

3189   \ifnum\MT@stretch=\m@ne
3190     \let\MT@stretch\MT@stretch@default
3191   \fi

```

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

```

3192   \ifnum\MT@shrink=\m@ne
3193     \ifnum\MT@stretch>\z@
3194       \let\MT@shrink\MT@stretch
3195     \else
3196       \let\MT@shrink\MT@shrink@default
3197   \fi
3198 \fi

```

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

```

3199   \ifnum\MT@step=\m@ne
3200     \ifnum\MT@stretch>\MT@shrink
3201       \ifnum\MT@shrink=\z@
3202         \@tempcnta=\MT@stretch
3203       \else
3204         \@tempcnta=\MT@shrink
3205       \fi
3206     \else
3207       \ifnum\MT@stretch=\z@
3208         \@tempcnta=\MT@shrink
3209       \else
3210         \@tempcnta=\MT@stretch
3211       \fi
3212     \fi
3213     \divide\@tempcnta 5\relax
3214   \else
3215     \@tempcnta=\MT@step
3216     \ifnum\@tempcnta=\z@
3217       \MT@warning@n1{The expansion step cannot be set to zero.\MessageBreak
3218         Setting it to one}
3219     \fi
3220   \fi
3221   \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3222   \edef\MT@step{\number\@tempcnta\space}

```

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

```

3223   \let\MT@auto\@empty
3224   \ifMT@auto
3225     \ifnum\MT@pdf@tex@no > \thr@@

```

We turn off automatic expansion if output mode is DVI.

```

3226     \ifnum\pdf@output=\z@
3227       \ifMT@opt@auto
3228         \MT@warning@n1{%

```

```

3229         Automatic font expansion only works for PDF output.\MessageBreak
3230         However, you are creating a DVI file. I will switch\MessageBreak
3231         automatic font expansion off and hope that expanded\MessageBreak
3232         fonts are available}
3233     \fi
3234     \MT@autofalse
3235 \else
3236     \def\MT@auto{autoexpand}
3237 \fi

```

Also, if pdf_T_EX is too old.

```

3238 \else
3239     \ifMT@opt@auto
3240     \MT@warning@nl{%
3241         The pdftex you are using is too old for automatic\MessageBreak
3242         font expansion. I will switch it off and hope that\MessageBreak
3243         expanded fonts are available on your system.\MessageBreak
3244         Install pdftex version 1.20 or newer}
3245     \fi
3246     \MT@autofalse
3247     \def\MT@auto{1000 }
3248 \fi

```

No automatic expansion.

```

3249 \else
3250     \ifnum\MT@pdftex@no < 4
3251     \def\MT@auto{1000 }
3252 \fi
3253 \fi

```

Choose the appropriate macro for selected expansion.

```

3254 \ifMT@selected
3255     \let\MT@set@ex@codes\MT@set@ex@codes@s
3256 \else
3257     \let\MT@set@ex@codes\MT@set@ex@codes@n
3258 \fi

```

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

```

3259 \ifnum\MT@stretch=\z@
3260     \ifnum\MT@shrink=\z@
3261     \MT@warning@nl{%
3262         Both the stretch and shrink limit are set to zero.\MessageBreak
3263         Disabling font expansion}
3264     \MT@expansionfalse
3265 \fi
3266 \fi
3267 \fi
3268 \ifMT@expansion
3269     \pdfadjustspacing\MT@ex@level
3270     \MT@info@nl{\ifMT@auto Automatic f\else F\fi ont expansion enabled
3271         (level \number\MT@ex@level),\MessageBreak
3272         stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3273         step: \number\MT@step, \ifMT@selected\else non-\fi selected}
3274     \MT@ifdefined@c\MT@ex@setname{%
3275         \MT@info@nl{Using expansion set '\MT@ex@setname'}%
3276     }{%
3277         \global\let\MT@ex@setname\MT@default@ex@set
3278         \MT@info@nl{Using default expansion set '\MT@ex@setname'}%
3279     }

```

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

```

3280 \CheckCommand*\showhyphens}[1]{%
3281 \setbox0\vbox{\color@begingroup\everypar{}\parfillskip\z@skip
3282 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3283 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

```

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

```

3284 \gdef\showhyphens#1{%
3285 \setbox0\vbox{%
3286 \color@begingroup
3287 \pdfadjustspacing\z@
3288 \everypar{}%
3289 \parfillskip\z@skip\hsize\maxdimen
3290 \normalfont
3291 \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ #1%
3292 \color@endgroup}}
3293 \else
3294 \let\MT@expansion\relax
3295 \MT@info@n1{No font expansion}
3296 \fi
3297 *beta
3298 \ifnum\MT@pdfTeX@no > 5
3299 \ifMT@spacing
3300 \pdfadjustinterwordglue\@ne

```

Warning if `\nonfrenchspacing` is active, since space factors will be ignored with `\pdfadjustinterwordglue>0`. Why 1500? Because some packages redefine `\frenchspacing`. See the c.t.t thread ‘`frenchspacing` with AMS packages and babel’, started by this message from Philipp Lehman: <ddtbaj\$rob\$1@online.de> on August 16, 2005.

```

3301 \AtBeginDocument{%
3302 \ifnum\scode'\. > 1500
3303 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
3304 \MT@warning@n1{%
3305 \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3306 interword spacing will disable it. You might want\MessageBreak
3307 to add '\string\microtypecontext{spacing=nonfrench}'\MessageBreak
3308 to your preamble}%
3309 }%
3310 \fi
3311 }
3312 \MT@info@n1{Adjustment of interword spacing enabled}
3313 \MT@ifdefinedc\MT@sp@setname{%
3314 \MT@info@n1{Using spacing set '\MT@sp@setname'}%
3315 }{%
3316 \global\let\MT@sp@setname\MT@default@sp@set
3317 \MT@info@n1{Using default spacing set '\MT@sp@setname'}%
3318 }
3319 \else
3320 \let\MT@spacing\relax
3321 \MT@info@n1{No adjustment of interword spacing}
3322 \fi
3323 \ifMT@kerning
3324 \pdfprependkern\@ne
3325 \pdfappendkern\@ne
3326 \MT@info@n1{Adjustment of character kerning enabled}
3327 \MT@ifdefinedc\MT@kn@setname{%
3328 \MT@info@n1{Using kerning set '\MT@kn@setname'}%

```

```

3329 }{%
3330   \global\let\MT@kn@setname\MT@default@kn@set
3331   \MT@info@n{Using default kerning set '\MT@kn@setname'}%
3332 }
3333 \else

```

We do *not* set \MT@kerning to \relax since it is also used by \textls.

```

3334 \MT@info@n{No adjustment of character kerning}
3335 \fi
3336 \ifnum\MT@letterspacing=\m@ne
3337   \let\MT@letterspacing\MT@letterspacing@default
3338 \fi

```

If pdfTeX is too old, we disable everything.

```

3339 \else
3340   \ifMT@spacing
3341     \MT@warning@n{Adjustment of interword spacing only works with\MessageBreak
3342       pdftex version 1.3x or newer. Switching it off}%
3343   \else
3344     \MT@info@n{No adjustment of interword spacing}
3345   \fi
3346   \MT@spacingfalse
3347   \let\MT@spacing\relax
3348   \ifMT@kerning
3349     \MT@warning@n{Character kerning only works with\MessageBreak
3350       pdftex version 1.3x or newer. Switching it off}%
3351   \else
3352     \MT@info@n{No adjustment of character kerning}
3353   \fi
3354   \MT@kerningfalse
3355   \let\MT@kerning\relax
3356 \fi

```

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

```

3357 \ifMT@babel
3358   \MT@info@n{Redefining babel's language switching commands}
3359   \let\MT@orig@select@language\select@language
3360   \def\select@language#1{%
3361     \MT@orig@select@language{#1}%
3362     \MT@ifdefined@n{MT@babel@#1}{%
3363       \MT@vinfo{Changing to language '#1' on line \the\inputlineno}%
3364       \expandafter\MT@exp@one@n\expandafter\microtypecontext
3365       \csname MT@babel@#1\endcsname
3366     }{%
3367       \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3368     }%
3369   }
3370   \let\MT@orig@foreign@language\foreign@language
3371   \def\foreign@language#1{%
3372     \MT@orig@foreign@language{#1}%
3373     \MT@ifdefined@n{MT@babel@#1}{%
3374       \MT@vinfo{Changing to context '#1' on line \the\inputlineno}%
3375       \expandafter\MT@exp@one@n\expandafter\microtypecontext
3376       \csname MT@babel@#1\endcsname
3377     }{%
3378       \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3379     }%
3380   }

```

Disable babel's active characters.

```

3381 \ifMT@kerning
3382 \AtBeginDocument{%
3383   \ifpackageloaded{babel}{%
3384     \@tempswa false
3385     \ifpackagewith{babel}{french}\@tempswatrue\relax
3386     \ifpackagewith{babel}{frenchb}\@tempswatrue\relax
3387     \ifpackagewith{babel}{francais}\@tempswatrue\relax
3388     \if@tempswa
3389       \NoAutoSpaceBeforeFDP
3390       \MT@warning@nl{Switching off French babel's active punctuation characters}%
3391     \fi
3392   }{}%
3393 }%
3394 \fi
3395 \fi
3396 </beta>
3397 </package>

```

That was that.

14 Configuration Files

Let's now write the font configuration files.

```

3398 <*config>
3399

```

The following characters must not appear verbatim:

```

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

```

Comma and equal sign must be guarded with braces ('{,}', '{=}'). Of course, numerical identification is possible in any case. Ligatures and `\mathchardef` symbols always have to be specified numerically.

14.1 Character Inheritance

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

```

3400 <*m-t>
3401 %%% -----
3402 %%% CHARACTER INHERITANCE
3403

```

14.1.1 OT1

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

```

3404 \DeclareCharacterInheritance
3405 { encoding = OT1 }
3406 { f = {011}, % ff
3407   i = {\i},
3408   j = {\j},
3409   O = {\O},
3410   o = {\o},
3411 }
3412

```

14.1.2 T1

Candidates here: 028 (‘fi’), 029 (‘fl’), 030 (‘ffi’), 031 (‘ffl’), 156 (‘IJ’ ligature), 188 (‘ij’), Æ, æ, Œ, œ.

```

3413 \DeclareCharacterInheritance
3414 { encoding = T1 }
3415 { A = {\'A,\'A,\^A,\~A,\"A,\r{A},\k{A},\u{A}},
3416   a = {\'a,\'a,\^a,\~a,\"a,\r{a},\k{a},\u{a}},
3417   C = {\'C,\c{C},\v{C}},
3418   c = {\'c,\c{c},\v{c}},
3419   D = {\v{D},\DH},
3420   d = {\v{d},\dj},
3421   E = {\'E,\'E,\^E,\~E,\k{E},\v{E}},
3422   e = {\'e,\'e,\^e,\~e,\k{e},\v{e}},
3423   f = {027}, % ff
3424   G = {\u{G}},
3425   g = {\u{g}},
3426   I = {\'I,\'I,\^I,\~I,\"I,\.I},
3427   i = {\'i,\'i,\^i,\~i,\"i,\i},
3428   j = {\j},
3429   L = {\L,\'L,\v{L}},
3430   l = {\l,\'l,\v{l}},
3431   N = {\'N,\~N,\v{N}},
3432   n = {\'n,\~n,\v{n}},
3433   O = {\O,\'O,\'O,\^O,\~O,\"O,\H{O}},
3434   o = {\o,\'o,\'o,\^o,\~o,\"o,\H{o}},
3435   R = {\'R,\v{R}},
3436   r = {\'r,\v{r}},
3437   S = {\'S,\c{S},\v{S},\SS},
3438   s = {\'s,\c{s},\v{s}},
3439   T = {\c{T},\v{T}},
3440   t = {\c{t},\v{t}},
3441   U = {\'U,\'U,\^U,\~U,\"U,\H{U},\r{U}},
3442   u = {\'u,\'u,\^u,\~u,\"u,\H{u},\r{u}},
3443   Y = {\'Y,\"Y},
3444   y = {\'y,\"y},
3445   Z = {\'Z,\.Z,\v{Z}},
3446   z = {\'z,\.z,\v{z}},
3447   - = {127},
3448 }
3449

```

14.1.3 LY1

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

```

3450 \DeclareCharacterInheritance
3451 { encoding = LY1 }

```

```

3452 { A = {\ 'A, \ 'A, \ ^A, \ ~A, \ "A, \ r{A}},
3453      a = {\ 'a, \ 'a, \ ^a, \ ~a, \ "a, \ r{a}},
3454      C = {\ c{C}},
3455      c = {\ c{c}},
3456      D = {\ DH},
3457      E = {\ 'E, \ 'E, \ ^E, \ "E},
3458      e = {\ 'e, \ 'e, \ ^e, \ "e},
3459      f = {011}, % ff
3460      I = {\ 'I, \ 'I, \ ^I, \ "I},
3461      i = {\ 'i, \ 'i, \ ^i, \ "i, \ i},
3462      L = {\ L},
3463      l = {\ l},
3464      N = {\ ~N},
3465      n = {\ ~n},
3466      O = {\ 'O, \ 'O, \ ^O, \ ~O, \ "O, \ O},
3467      o = {\ 'o, \ 'o, \ ^o, \ ~o, \ "o, \ o},
3468      S = {\ v{S}},
3469      s = {\ v{s}},
3470      U = {\ 'U, \ 'U, \ ^U, \ "U},
3471      u = {\ 'u, \ 'u, \ ^u, \ "u},
3472      Y = {\ 'Y, \ "Y},
3473      y = {\ 'y, \ "y},
3474      Z = {\ v{Z}},
3475      z = {\ v{z}},
3476 }
3477

```

14.1.4 OT4

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

```

3478 \DeclareCharacterInheritance
3479 { encoding = OT4 }
3480 { A = {\ k{A}},
3481      a = {\ k{a}},
3482      C = {\ 'C},
3483      c = {\ 'c},
3484      E = {\ k{E}},
3485      e = {\ k{e}},
3486      f = {011}, % ff
3487      i = {\ i},
3488      j = {\ j},
3489      L = {\ L},
3490      l = {\ l},
3491      N = {\ 'N},
3492      n = {\ 'n},
3493      O = {\ O, \ 'O},
3494      o = {\ o, \ 'o},
3495      S = {\ 'S},
3496      s = {\ 's},
3497      Z = {\ 'Z, \ .Z},
3498      z = {\ 'z, \ .z},
3499 }
3500

```


14.1.5 T5

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

```

3501 \DeclareCharacterInheritance
3502   { encoding = T5 }
3503   { A = {\A,\A,\A,\h{A},\d{A},\^A,\u{A},
3504         \'\Acircumflex,\'\Acircumflex,\~\Acircumflex,\h\Acircumflex,\d\Acircumflex,
3505         \'\Abreve,\'\Abreve,\~\Abreve,\h\Abreve,\d\Abreve},
3506     a = {\a,\a,\a,\h{a},\d{a},\^a,\u{a},
3507         \'\acircumflex,\'\acircumflex,\~\acircumflex,\h\acircumflex,\d\acircumflex,
3508         \'\abreve,\'\abreve,\~\abreve,\h\abreve,\d\abreve},
3509     D = {\DJ},
3510     d = {\dj},
3511     E = {\E,\E,\E,\h{E},\d{E},\^E,
3512         \'\Ecircumflex,\'\Ecircumflex,\~\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
3513     e = {\e,\e,\e,\h{e},\d{e},\^e,
3514         \'\ecircumflex,\'\ecircumflex,\~\ecircumflex,\h\ecircumflex,\d\ecircumflex},
3515     I = {\I,\I,\I,\h{I},\d{I}},
3516     i = {\i,\i,\i,\h{i},\d{i},\i},
3517     O = {\O,\O,\O,\h{O},\d{O},\^O,\horn{O},
3518         \'\Ocircumflex,\'\Ocircumflex,\~\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
3519         \'\Ohorn,\'\Ohorn,\~\Ohorn,\h\Ohorn,\d\Ohorn},
3520     o = {\o,\o,\o,\h{o},\d{o},\^o,\horn{o},
3521         \'\ocircumflex,\'\ocircumflex,\~\ocircumflex,\h\ocircumflex,\d\ocircumflex,
3522         \'\ohorn,\'\ohorn,\~\ohorn,\h\ohorn,\d\ohorn},
3523     U = {\U,\U,\U,\h{U},\d{U},\horn{U},
3524         \'\Uhorn,\'\Uhorn,\~\Uhorn,\h\Uhorn,\d\Uhorn},
3525     u = {\u,\u,\u,\h{u},\d{u},\horn{u},
3526         \'\uhorn,\'\uhorn,\~\uhorn,\h\uhorn,\d\uhorn},
3527     Y = {\Y,\Y,\Y,\h{Y},\d{Y}},
3528     y = {\y,\y,\y,\h{y},\d{y}},
3529   }
3530
3531 </m-t>

```

14.2 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).

```

3532 <*m-t>
3533 %%% -----
3534 %%% EXPANSION SETTINGS
3535
3536 \SetExpansion
3537 <m-t> [ name      = default      ]
3538 <bch>  [ name      = bch-default  ]
3539 <cmr>  [ name      = cmr-default  ]
3540 <pad>  [ name      = pad-default  ]
3541 <pmn>  [ name      = pmn-default  ]
3542 <ppl>  [ name      = ppl-default  ]
3543 <ptm>  [ name      = ptm-default  ]
3544 <m-t>  { encoding = {OT1,OT4,T1,LY1} }
3545 <!m-t> { encoding = {OT1,T1,LY1},
3546 <bch>   family    = bch }
3547 <cmr>   family    = cmr }
3548 <pad>   family    = {pad,padx,padj} }

```

```

3549 <pmn>    family = {pmn,pmnx,pmnj} }
3550 <ppl>     family = {ppl,pplx,pplj} }
3551 <ptm>     family = {ptm,ptmx,ptmj} }
3552 {
3553     A = 500,      a = 700,
3554     \AE = 500,    \ae = 700,
3555     B = 700,      b = 700,
3556     C = 700,      c = 700,
3557     D = 500,      d = 700,
3558     E = 700,      e = 700,
3559     F = 700,
3560     G = 500,      g = 700,
3561     H = 700,      h = 700,
3562     K = 700,      k = 700,
3563     M = 700,      m = 700,
3564     N = 700,      n = 700,
3565     O = 500,      o = 700,
3566     \OE = 500,    \oe = 700,
3567     P = 700,      p = 700,
3568     Q = 500,      q = 700,
3569     R = 700,
3570     S = 700,      s = 700,
3571     U = 700,      u = 700,
3572     W = 700,      w = 700,
3573     Z = 700,      z = 700,
3574     2 = 700,
3575     3 = 700,
3576     6 = 700,
3577     8 = 700,
3578     9 = 700,
3579 }
3580

```

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

```

3581 \SetExpansion
3582 [ name = T5 ]
3583 { encoding = T5 }
3584 {
3585     A = 500,      a = 700,
3586     B = 700,      b = 700,
3587     C = 700,      c = 700,
3588     D = 500,      d = 700,
3589     E = 700,      e = 700,
3590     F = 700,
3591     G = 500,      g = 700,
3592     H = 700,      h = 700,
3593     K = 700,      k = 700,
3594     M = 700,      m = 700,
3595     N = 700,      n = 700,
3596     O = 500,      o = 700,
3597     P = 700,      p = 700,
3598     Q = 500,      q = 700,
3599     R = 700,
3600     S = 700,      s = 700,
3601     U = 700,      u = 700,
3602     W = 700,      w = 700,
3603     Z = 700,      z = 700,
3604     2 = 700,
3605     3 = 700,
3606     6 = 700,

```

```

3607      8 = 700,
3608      9 = 700,
3609    }
3610
3611 /m-t

```

14.3 Character Protrusion

```

3612 %% -----
3613 %% PROTRUSION SETTINGS
3614

```

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

```

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

```

We also create configuration files for the fonts Bitstream Charter (NFSS code `bch`), Computer Modern Roman (`cmr`), Palatino (`ppl`, `pplx`, `pplj`), Times (`ptm`, `ptmx`, `ptmj`), Adobe Garamond (`pad`, `padx`, `padj`) and Minion¹⁷ (`pmnx`, `pmnj`), and for the AMS math fonts (`msa`, `msb`, `euf`, `eus`).

14.3.1 Default

The default settings always use the most moderate value.

```

3615 !*cfg-u
3616 \SetProtrusion
3617 /m-t [ name      = default ]

```

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

```

3618 <bch> [ name = bch-default ]
3619 <cmr> [ name = cmr-default ]
3620 <pad> [ name = pad-default ]
3621 <pmn> [ name = pmnj-default ]
3622 <ppl> [ name = ppl-default ]
3623 <ptm> [ name = ptm-default ]
3624 <m-t> { encoding = OT1 }
3625 <cmr> { }
3626 <bch|pad|pmn> { encoding = OT1,
3627 <ppl|ptm> { encoding = {OT1,OT4},
3628 <bch> family = bch }
3629 <pad> family = {pad,padx,padj} }
3630 <pmn> family = pmnj }
3631 <ppl> family = {ppl,pplx,pplj} }
3632 <ptm> family = {ptm,ptmx,ptmj} }
3633 {
3634 A = {50,50},
3635 <m-t|pad|ptm> \AE = {50, },
3636 <bch|pad|pmn> C = {50, },
3637 <bch|pad|pmn> D = { ,50},
3638 <m-t|bch|cmr|pad|pmn|ptm> F = { ,50},
3639 <bch|pad|pmn> G = {50, },
3640 <m-t|cmr|pad|pmn|ppl|ptm> J = {50, },
3641 <bch> J = {100, },
3642 K = { ,50},
3643 <m-t|bch|cmr|pad|pmn|ppl> L = { ,50},
3644 <ptm> L = { ,80},
3645 <bch|pad|pmn> O = {50,50},
3646 <bch|pad|pmn> \OE = {50, },
3647 <bch|pad|pmn> Q = {50,70},
3648 <bch> R = { ,50},
3649 T = {50,50},
3650 V = {50,50},
3651 W = {50,50},
3652 X = {50,50},
3653 <m-t|bch|cmr|pad|pmn|ppl> Y = {50,50},
3654 <ptm> Y = {80,80},
3655 k = { ,50},
3656 <pmn> l = { , -50},
3657 <pad|ppl> p = {50,50},
3658 <pad|ppl> q = {50, },
3659 r = { ,50},
3660 <cmr|pad|pmn> t = { ,70},
3661 <bch> t = { ,50},
3662 v = {50,50},
3663 w = {50,50},
3664 x = {50,50},
3665 <m-t|bch|pad|pmn> y = { ,50},
3666 <cmr|ppl|ptm> y = {50,70},

3667 <cmr> 0 = { ,50},
3668 <m-t> 1 = {50,50},
3669 <bch|pad|ptm> 1 = {150,150},
3670 <cmr> 1 = {100,200},
3671 <pmn> 1 = { ,50},
3672 <ppl> 1 = {100,100},
3673 <bch|cmr|pad> 2 = {50,50},
3674 <cmr|pad> 3 = {50,50},
3675 <bch|pmn> 3 = {50, },
3676 <m-t|pad> 4 = {50,50},

```

```

3677 <bch> 4 = {100,50},
3678 <cmr> 4 = {70,70},
3679 <pmn> 4 = {50, },
3680 <ptm> 4 = {70, },
3681 <cmr> 5 = { ,50},
3682 <pad> 5 = {50,50},
3683 <bch> 6 = {50, },
3684 <cmr> 6 = { ,50},
3685 <pad> 6 = {50,50},
3686 <m-t> 7 = {50,50},
3687 <bch|pad|pmn> 7 = {50,80},
3688 <cmr|ptm> 7 = {50,100},
3689 <ppl> 7 = { ,50},
3690 <cmr> 8 = { ,50},
3691 <bch|pad> 9 = {50,50},
3692 <cmr> 9 = { ,50},
3693 <m-t|cmr|pad|pmn|ppl|ptm> . = { ,700},
3694 <bch> . = { ,600},
3695 {,}= { ,500},
3696 <m-t|cmr|pad|pmn|ppl|ptm> : = { ,500},
3697 <bch> : = { ,400},
3698 <m-t|bch|pad|pmn|ptm> ; = { ,300},
3699 <cmr|ppl> ; = { ,500},
3700 ! = { ,100},
3701 <m-t|pad|pmn|ptm> ? = { ,100},
3702 <bch|cmr|ppl> ? = { ,200},
3703 <pmn> " = {300,300},
3704 <m-t|bch|cmr|pad|pmn|ppl> @ = {50,50},
3705 <ptm> @ = {100,100},
3706 ~ = {200,250},
3707 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
3708 <cmr> _ = {200,200},
3709 <pad|ppl|ptm> & = {50,100},
3710 <m-t|cmr|pad|pmn> \% = {50,50},
3711 <bch> \% = { ,50},
3712 <ppl|ptm> \% = {100,100},
3713 <m-t|ppl|ptm> * = {200,200},
3714 <bch|pmn> * = {200,300},
3715 <cmr|pad> * = {300,300},
3716 <m-t|cmr|ppl|ptm> + = {250,250},
3717 <bch> + = {150,250},
3718 <pad> + = {300,300},
3719 <pmn> + = {150,200},
3720 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
3721 <bch> ( = {200, }, ) = { ,200},
3722 <cmr|ppl> ( = {100, }, ) = { ,300},
3723 <bch|pmn> [ = {100, }, ] = { ,100},

3724 <m-t|pad|pmn|ptm> / = {100,200},
3725 <bch> / = { ,200},
3726 <cmr|ppl> / = {200,300},
3727 <m-t|ptm> - = {500,500},
3728 <bch|cmr|ppl> - = {400,500},
3729 <pad> - = {300,500},
3730 <pmn> - = {200,400},
3731 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
3732 <bch> \textendash = {200,300}, \textendash = {150,250},
3733 <cmr> \textendash = {400,300}, \textendash = {300,200},
3734 <pad|ppl|ptm> \textendash = {300,300}, \textendash = {200,200},

```

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

```

3735 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
3736 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
3737 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
3738 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
3739 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300},
3740 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600},
3741 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400},
3742 }
3743

```

Greek uppercase letters are in OT1 encoding only.

```

3744 <*cmr>
3745 \SetProtrusion
3746 [ name = cmr-OT1,
3747   load = cmr-default ]
3748 { encoding = {OT1,OT4},
3749   family = cmr }
3750 {
3751   \AE = { 50, },
3752   "00 = { ,150}, % \Gamma
3753   "01 = {100,100}, % \Delta
3754   "02 = { 50, 50}, % \Theta
3755   "03 = {100,100}, % \Lambda
3756 % "04 = { , }, % \Xi
3757 % "05 = { , }, % \Pi
3758   "06 = { 50, 50}, % \Sigma
3759   "07 = {100,100}, % \Upsilon
3760   "08 = { 50, 50}, % \Phi
3761   "09 = { 50, 50}, % \Psi
3762 % "0A = { , }, % \Omega
3763 }
3764
3765 </cmr>

```

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

```

3766 \SetProtrusion
3767 <m-t> [ name = T1-default,
3768 <bch> [ name = bch-T1,
3769 <cmr> [ name = cmr-T1,
3770 <pad> [ name = pad-T1,
3771 <pmn> [ name = pmnj-T1,
3772 <ppl> [ name = ppl-T1,
3773 <ptm> [ name = ptm-T1,
3774 <m-t> load = default ]
3775 <bch> load = bch-default ]
3776 <cmr> load = cmr-default ]
3777 <pad> load = pad-default ]
3778 <pmn> load = pmnj-default ]
3779 <ppl> load = ppl-default ]
3780 <ptm> load = ptm-default ]
3781 <m-t> { encoding = {T1,LY1} }
3782 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
3783 <ptm> { encoding = {T1},
3784 <bch> family = bch }
3785 <cmr> family = cmr }
3786 <pad> family = {pad,padx,padj} }

```

```

3787 <pmn>    family = pmnj }
3788 <ppl>    family = {ppl,pplx,pplj} }
3789 <ptm>    family = {ptm,ptmx,ptmj} }
3790 {
3791 <cmr>    \AE = {50, },
3792 <pmn>    \TH = { ,50},
3793 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
3794 <bch>    \textbackslash = {150,200},
3795 <cmr|ppl> \textbackslash = {200,300},
3796 <cmr>    \textquotedblleft = {200,600},
3797 <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.

```

3798 <m-t|cmr|pad|ppl|ptm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3799 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
3800 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
3801 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
3802 <m-t>    \guillemotleft = {200,200}, \guillemotright = {200,200},
3803 <cmr>    \guillemotleft = {300,200}, \guillemotright = {100,400},
3804 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
3805 <pad>    \guillemotleft = {300,300}, \guillemotright = {200,400},
3806 <ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
3807 <m-t|bch|cmr|pad|pmn|ppl> \textexclamdown = {100, }, \textquestiondown = {100, },
3808 <ptm>    \textexclamdown = {200, }, \textquestiondown = {200, },
3809 <m-t|cmr|pad|ppl|ptm> \textbraceleft = {400,200}, \textbraceright = {200,400},
3810 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
3811 <m-t|bch|cmr|pad|ppl|ptm> \textless = {200,100}, \textgreater = {100,200},
3812 <pmn>    \textless = {100, }, \textgreater = { ,100},
3813 <pmn>    \textvisiblespace = {100,100}, % not in LY1
3814 % \dh = { , },
3815 % \th = { , },
3816 % \NG = { , },
3817 % \ng = { , },
3818 % \textasciicircum = { , },
3819 % \textbar = { , },
3820 % \textsterling = { , }, % also in TS1
3821 % \textsection = { , }, % also in TS1
3822 }
3823
3824 <*cmr>

```

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.

```

3825 \SetProtrusion
3826 [ name = cmr-T5,
3827   load = cmr-default ]
3828 { encoding = T5,
3829   family = cmr }
3830 {
3831   \textbackslash = {200,300},
3832   \textquotedblleft = {200,600},
3833   \textquotedbl = {300,300},
3834   \quotesinglbase = {400,400}, \quotedblbase = {400,400},
3835   \guilsinglleft = {400,400}, \guilsinglright = {300,500},
3836   \guillemotleft = {300,200}, \guillemotright = {100,400},
3837   \textbraceleft = {400,200}, \textbraceright = {200,400},
3838   \textless = {200,100}, \textgreater = {100,200},
3839 }

```

3840

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

```

3841 \SetProtrusion
3842 [ name      = lmr-T1,
3843   load      = cmr-T1 ]
3844 { encoding = {T1,LY1},
3845   family   = lmr      }
3846 {
3847   \textquotedblleft = {500,300},
3848   \quotedblbase     = {500,300},
3849 }
3850
3851 /cmr
3852 *pmn
3853 \SetProtrusion
3854 [ name      = pmnx-OT1,
3855   load      = pmnj-default ]
3856 { encoding = OT1,
3857   family   = pmnx }
3858 {
3859   1 = {230,180},
3860 }
3861
3862 \SetProtrusion
3863 [ name      = pmnx-T1,
3864   load      = pmnj-T1 ]
3865 { encoding = {T1,LY1},
3866   family   = pmnx }
3867 {
3868   1 = {230,180},
3869 }
3870
3871 /pmn

```

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

```

3872 *ptm
3873 \SetProtrusion
3874 [ name      = ptm-LY1,
3875   load      = ptm-T1 ]
3876 { encoding = LY1,
3877   family   = {ptm,ptmx,ptmj} }
3878 {
3879   \texttrademark      = {100,100},
3880   \textregistered     = {100,100},
3881   \textcopyright      = {100,100},
3882   \textdegree         = {300,300},
3883   \textminus          = {200,200},
3884   \textellipsis       = {100,100},
3885   \texteuro           = { , }, % ?
3886   \textcent           = {100,100},
3887   \textquotesingle    = {500,500},
3888   \textflorin         = { 50, 70},
3889   \textdagger         = {150,150},
3890   \textdaggerdbl      = {100,100},
3891   \textperthousand    = { , 50},
3892   \textbullet         = {150,150},

```



```

3893 \textonesuperior      = {100,100},
3894 \texttwosuperior      = { 50, 50},
3895 \textthreesuperior    = { 50, 50},
3896 \textperiodcentered   = {300,300},
3897 \textplusminus        = { 50, 80},
3898 \textmultiply         = {100,100},
3899 \textdivide          = { 50,150},
3900 % \textbrokenbar      = { , },
3901 % \textyen            = { , },
3902 % \textfractionsolidus = { , },
3903 % \textordfeminine     = { , },
3904 % \textordmasculine    = { , },
3905 % \textmu             = { , },
3906 % \textparagraph      = { , },
3907 % \textonequarter     = { , },
3908 % \textonehalf        = { , },
3909 % \textthreequarters   = { , },
3910 }
3911
3912 </ptm>

```

14.3.2 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.

```

3913 \SetProtrusion
3914 <m-t> [ name = OT1-it ]
3915 <bch> [ name = bch-it ]
3916 <cmr> [ name = cmr-it ]
3917 <pad> [ name = pad-it ]
3918 <pmn> [ name = pmnj-it ]
3919 <ppl> [ name = ppl-it ]
3920 <ptm> [ name = ptm-it ]
3921 <m-t|bch|pad|pmn> { encoding = OT1,
3922 <cmr> { }
3923 <ppl|ptm> { encoding = {OT1,OT4},
3924 <bch> family = bch,
3925 <pad> family = {pad,padx,padj},
3926 <pmn> family = pmnj,
3927 <ppl> family = {ppl,pplx,pplj},
3928 <ptm> family = {ptm,ptmx,ptmj},
3929 <!cmr> shape = {it,sl} }
3930 {
3931 <cmr|ptm> A = {100,50},
3932 <pad|pmn> A = {50, },
3933 <ppl> A = {50,50},
3934 <ptm> \AE = {100, },
3935 <pad|ppl> \AE = {50, },
3936 <pmn> \AE = { , -50},
3937 <cmr|pad|ppl|ptm> B = {50, },
3938 <pmn> B = {20, -50},
3939 <bch|ppl|ptm> C = {50, },
3940 <cmr|pad> C = {100, },
3941 <pmn> C = {50, -50},
3942 <cmr|pad|ppl|ptm> D = {50,50},
3943 <pmn> D = {20, },
3944 <cmr|pad|ppl|ptm> E = {50, },

```

```

3945 <pmn>      E = {20,-50},
3946 <cmr|pad|ptm> F = {100, },
3947 <pmn>      F = {10,  },
3948 <ppl>      F = {50,  },
3949 <bch|ppl|ptm> G = {50,  },
3950 <cmr|pad>   G = {100, },
3951 <pmn>      G = {50,-50},
3952 <cmr|pad|ppl|ptm> H = {50,  },
3953 <cmr|pad|ptm> I = {50,  },
3954 <pmn>      I = {20,-50},
3955 <cmr|ptm>   J = {100, },
3956 <pad>      J = {50,  },
3957 <pmn>      J = {20,  },
3958 <cmr|pad|ppl|ptm> K = {50,  },
3959 <pmn>      K = {20,  },
3960 <cmr|pad|ppl|ptm> L = {50,  },
3961 <pmn>      L = {20,50},
3962 <cmr|ptm>   M = {50,  },
3963 <pmn>      M = { , -30},
3964 <cmr|ptm>   N = {50,  },
3965 <pmn>      N = { , -30},
3966 <bch|pmn|ppl|ptm> O = {50,  },
3967 <cmr|pad>   O = {100, },
3968 <bch|pmn|ppl|ptm> \OE = {50,  },
3969 <pad>      \OE = {100, },
3970 <cmr|pad|ppl|ptm> P = {50,  },
3971 <pmn>      P = {20,-50},
3972 <bch|pmn|ppl|ptm> Q = {50,  },
3973 <cmr|pad>   Q = {100, },
3974 <cmr|pad|ppl|ptm> R = {50,  },
3975 <pmn>      R = {20,  },
3976 <bch|cmr|pad|ppl|ptm> S = {50,  },
3977 <pmn>      S = {20,-30},
3978 <bch|cmr|pad|ppl|ptm> $ = {50,  },
3979 <pmn>      $ = {20,-30},
3980 <bch|pmn>   T = {70,  },
3981 <cmr|pad|ppl|ptm> T = {100, },
3982 <cmr|pad|ppl|ptm> U = {50,  },
3983 <pmn>      U = {50,-50},
3984 <cmr|pad|pmn> V = {100, },
3985 <ppl|ptm>   V = {100,50},
3986 <cmr|pad|pmn> W = {100, },
3987 <ppl>      W = {50,  },
3988 <ptm>      W = {100,50},
3989 <cmr|ppl|ptm> X = {50,  },
3990 <cmr|ptm>   Y = {100, },
3991 <pmn>      Y = {50,  },
3992 <ppl>      Y = {100,50},
3993 <pmn>      Z = { , -50},
3994 <pmn>      d = { , -50},
3995 <pad|pmn>   f = { , -100},
3996 <pmn>      i = { , -30},
3997 <pmn>      j = { , -30},
3998 <pmn>      l = { , -100},
3999 <bch>      o = {50,50},
4000 <bch>      p = { , 50},
4001 <pmn>      p = {-50,  },
4002 <bch>      q = {50,  },
4003 <pmn>      r = { , 50},
4004 <bch>      t = { , 50},

```

```

4005 <pmn>      v = {50, },
4006 <bch>      w = { ,50},
4007 <pmn>      w = {50, },
4008 <bch>      y = { ,50},
4009 <cmr>      0 = {100, },
4010 <bch|ptm>  1 = {150,100},
4011 <cmr>      1 = {200,50},
4012 <pad>      1 = {150, },
4013 <pmn>      1 = {50, },
4014 <ppl>      1 = {100, },
4015 <cmr>      2 = {100,-100},
4016 <pad|ppl|ptm> 2 = {50, },
4017 <pmn>      2 = {-50, },
4018 <bch>      3 = {50, },
4019 <cmr>      3 = {100,-100},
4020 <pmn>      3 = {-100, },
4021 <ptm>      3 = {100,50},
4022 <bch>      4 = {100, },
4023 <cmr|pad>  4 = {150, },
4024 <ppl|ptm>  4 = {50, },
4025 <cmr>      5 = {100, },
4026 <ptm>      5 = {50, },
4027 <bch>      6 = {50, },
4028 <cmr>      6 = {100, },
4029 <bch|pad|ptm> 7 = {100, },
4030 <cmr>      7 = {200,-150},
4031 <pmn>      7 = {20, },
4032 <ppl>      7 = {50, },
4033 <cmr>      8 = {50,-50},
4034 <cmr>      9 = {100,-100},
4035 <m-t|cmr|pad|pmn|ppl> . = { ,500},
4036 <bch|ptm>   . = { ,700},
4037 <m-t|cmr|pad|pmn|ppl> {,}= { ,500},
4038 <bch>      {,}= { ,600},
4039 <ptm>      {,}= { ,700},
4040 <m-t|cmr|pad|ppl> : = { ,300},
4041 <bch>      : = { ,400},
4042 <pmn>      : = { ,200},
4043 <ptm>      : = { ,500},
4044 <m-t|cmr|pad|ppl> ; = { ,300},
4045 <bch>      ; = { ,400},
4046 <pmn>      ; = { ,200},
4047 <ptm>      ; = { ,500},
4048 <ptm>      ! = { ,100},
4049 <bch>      ? = { ,200},
4050 <ptm>      ? = { ,100},
4051 <ppl>      ? = { ,300},
4052 <pmn>      " = {400,200},
4053 <m-t|bch|pmn> _ = { ,100},
4054 <cmr>      _ = {100,200},
4055 <pad|ppl|ptm> _ = {100,100},
4056 <m-t|pad|pmn|ppl|ptm> & = {50,50},
4057 <bch>      & = { ,80},
4058 <cmr>      & = {100,50},
4059 <m-t|cmr|pad|pmn> \% = {100, },
4060 <bch>      \% = {50,50},
4061 <ppl|ptm>   \% = {100,100},
4062 <m-t|pmn|ppl> * = {200,200},
4063 <bch>      * = {300,200},
4064 <cmr>      * = {400,100},

```

```

4065 <pad>      * = {500,100},
4066 <ptm>      * = {400,200},
4067 <m-t|cmr|pmn|ppl>      + = {150,200},
4068 <bch>      + = {250,250},
4069 <pad|ptm>      + = {250,200},
4070 <m-t|pad|pmn|ppl>      @ = {50,50},
4071 <bch>      @ = {80,50},
4072 <cmr>      @ = {200,50},
4073 <ptm>      @ = {150,150},
4074 <m-t|bch>      ~ = {150,150},
4075 <cmr|pad|pmn|ppl|ptm>      ~ = {200,150},
4076      ( = {200, }, ) = { ,200},
4077 <m-t|cmr|pad|ppl|ptm>      / = {100,200},
4078 <bch>      / = { ,150},
4079 <pmn>      / = {100,150},
4080 <m-t>      - = {300,300},
4081 <bch|pad>      - = {300,400},
4082 <pmn>      - = {200,300},
4083 <cmr>      - = {500,300},
4084 <ppl>      - = {300,500},
4085 <ptm>      - = {500,500},
4086 <m-t|pmn>      \textendash = {200,200}, \textendash = {150,150},
4087 <bch>      \textendash = {200,300}, \textendash = {150,200},
4088 <cmr>      \textendash = {500,300}, \textendash = {400,200},
4089 <pad|ppl|ptm>      \textendash = {300,300}, \textendash = {200,200},
4090 <m-t|bch|pmn>      \textquoteleft = {400,200}, \textquoteright = {400,200},
4091 <cmr|pad>      \textquoteleft = {800,200}, \textquoteright = {800,200},
4092 <ppl>      \textquoteleft = {700,400}, \textquoteright = {700,400},
4093 <ptm>      \textquoteleft = {800,500}, \textquoteright = {800,500},
4094 <m-t|bch|pmn>      \textquotedblleft = {400,200}, \textquotedblright = {400,200},
4095 <cmr>      \textquotedblleft = {700,100}, \textquotedblright = {500,300},
4096 <pad>      \textquotedblleft = {700,200}, \textquotedblright = {700,200},
4097 <ppl>      \textquotedblleft = {500,300}, \textquotedblright = {500,300},
4098 <ptm>      \textquotedblleft = {700,400}, \textquotedblright = {700,400},
4099     }
4100
4101 <*cmr>
4102 \SetProtrusion
4103 [ name = cmr-it-OT1,
4104   load = cmr-it ]
4105 { encoding = {OT1,OT4},
4106   family = cmr,
4107   shape = it      }
4108 {
4109   \AE = {100, },
4110   \OE = {100, },
4111   "00 = {200,150}, % \Gamma
4112   "01 = {150,100}, % \Delta
4113   "02 = {150, 50}, % \Theta
4114   "03 = {150, 50}, % \Lambda
4115   "04 = {100,100}, % \Xi
4116   "05 = {100,100}, % \Pi
4117   "06 = {100, 50}, % \Sigma
4118   "07 = {200,150}, % \Upsilon
4119   "08 = {150, 50}, % \Phi
4120   "09 = {150,100}, % \Psi
4121   "0A = { 50, 50}, % \Omega
4122 }
4123
4124 </cmr>

```

```

4125 \SetProtrusion
4126 <m-t> [ name = T1-it-default,
4127 <bch> [ name = bch-it-T1,
4128 <cmr> [ name = cmr-it-T1,
4129 <pad> [ name = pad-it-T1,
4130 <pmn> [ name = pmnj-it-T1,
4131 <ppl> [ name = ppl-it-T1,
4132 <ptm> [ name = ptm-it-T1,
4133 <m-t> load = OT1-it ]
4134 <bch> load = bch-it ]
4135 <cmr> load = cmr-it ]
4136 <pmn> load = pmnj-it ]
4137 <pad> load = pad-it ]
4138 <ppl> load = ppl-it ]
4139 <ptm> load = ptm-it ]
4140 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4141 <ptm> { encoding = {T1},
4142 <bch> family = bch,
4143 <cmr> family = cmr,
4144 <pmn> family = pmnj,
4145 <pad> family = {pad,padx,padj},
4146 <ppl> family = {ppl,pplx,pplj},
4147 <ptm> family = {ptm,ptmx,ptmj},
4148 <!cmr> shape = {it,sl} }
4149 <cmr> shape = it }
4150 {
4151 <cmr> \AE = {100, },
4152 <cmr> \OE = {100, },
4153 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
4154 <cmr> \textbackslash = {300,300},
4155 <bch> \textbackslash = {150,150},
4156 <pmn> \textbackslash = {100,150},
4157 <pmn> 031 = { , -100}, % ffi
4158 <cmr|ptm> 156 = {100, }, % IJ
4159 <pad> 156 = {50, }, % IJ
4160 <pmn> 156 = {20, }, % IJ
4161 <pmn> 188 = { , -30}, % ij
4162 <pmn> \v{t} = { ,100},
4163 <cmr> \textquotedblleft = {500,300},
4164 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
4165 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4166 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
4167 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
4168 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4169 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
4170 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4171 <pad> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
4172 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
4173 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
4174 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
4175 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
4176 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
4177 <m-t|pad|ppl> \textexclamdown = {100, }, \textquestiondown = {200, },
4178 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4179 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
4180 <m-t|ppl> \textbraceleft = {200,100}, \textbraceright = {200,200},
4181 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
4182 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
4183 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
4184 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100},

```

```

4185 <pmn> \textvisiblespace = {100,100},
4186 }
4187
4188 <*cmr>
4189 \SetProtrusion
4190 [ name = cmr-it-T5,
4191   load = cmr-it ]
4192 { encoding = T5,
4193   family = cmr,
4194   shape = it }
4195 {
4196   \textbackslash = {300,300},
4197   \quotesinglbase = {300,700}, \quotedblbase = {200,600},
4198   \guilsinglleft = {500,300}, \guilsinglright = {400,400},
4199   \guillemotleft = {400,100}, \guillemotright = {200,300},
4200   \textbraceleft = {400,100}, \textbraceright = {200,200},
4201   \textless = {300,100}, \textgreater = {200,100},
4202 }
4203

```

Slanted is very similar to italic.

```

4204 \SetProtrusion
4205 [ name = cmr-sl,
4206   load = cmr-it-OT1 ]
4207 { encoding = {OT1,OT4},
4208   family = cmr,
4209   shape = sl }
4210 {
4211   L = { ,50},
4212   f = { ,-50},
4213   - = {300, },
4214   \textendash = {400, }, \textemdash = {300, },
4215 }
4216
4217 \SetProtrusion
4218 [ name = cmr-sl-T1,
4219   load = cmr-it-T1 ]
4220 { encoding = {T1,LY1},
4221   family = cmr,
4222   shape = sl }
4223 {
4224   L = { ,50},
4225   f = { ,-50},
4226   - = {300, },
4227   \textendash = {400, }, \textemdash = {300, },
4228 }
4229
4230 \SetProtrusion
4231 [ name = cmr-sl-T5,
4232   load = cmr-it-T5 ]
4233 { encoding = T5,
4234   family = cmr,
4235   shape = sl }
4236 {
4237   L = { ,50},
4238   f = { ,-50},
4239   - = {300, },
4240   \textendash = {400, }, \textemdash = {300, },
4241 }
4242

```

```

4243 \SetProtrusion
4244 [ name = lmr-it-T1,
4245   load = cmr-it-T1 ]
4246 { encoding = {T1,LV1},
4247   family = lmr,
4248   shape = {it,sl} }
4249 {
4250   \textquotedblleft = {700,100},
4251   \quotedblbase = {600,300},
4252 }
4253

```

Oldstyle numerals are slightly different.

```

4254 \SetProtrusion
4255 [ name = cmr(oldstyle)-it,
4256   load = cmr-it-T1 ]
4257 { encoding = T1,
4258   family = {hfor,cmor},
4259   shape = {it,sl} }
4260 {
4261   1 = {250, 50},
4262   2 = {150,-100},
4263   3 = {100,-50},
4264   4 = {150,150},
4265   6 = {200,  },
4266   7 = {200, 50},
4267   8 = {150,-50},
4268   9 = {100, 50},
4269 }
4270
4271 </cmr>
4272 < *pmn>
4273 \SetProtrusion
4274 [ name = pmnx-it,
4275   load = pmnj-it ]
4276 { encoding = OT1,
4277   family = pmnx,
4278   shape = {it,sl} }
4279 {
4280   1 = {100,150},
4281 }
4282
4283 \SetProtrusion
4284 [ name = pmnx-it-T1,
4285   load = pmnj-it-T1 ]
4286 { encoding = {T1,LV1},
4287   family = pmnx,
4288   shape = {it,sl} }
4289 {
4290   1 = {100,150},
4291 }
4292
4293 </pmn>
4294 < *ptm>
4295 \SetProtrusion
4296 [ name = ptm-it-LV1,
4297   load = ptm-it-T1 ]
4298 { encoding = {LV1},
4299   family = {ptm,ptmx,ptmj},
4300   shape = {it,sl} }

```

```

4301 {
4302     \texttrademark      = {100,100},
4303     \textregistered     = {100,100},
4304     \textcopyright      = {100,100},
4305     \textdegree         = {300,100},
4306     \textminus          = {200,200},
4307     \textellipsis       = {100,150},
4308     \texteuro           = {  ,  },
4309     \textcent           = {100,100},
4310     \textquotesingle    = {500,  },
4311     \textflorin         = {100, 70},
4312     \textdagger         = {150,150},
4313     \textdaggerdbl      = {100,100},
4314     \textbullet         = {150,150},
4315     \textonesuperior    = {150,100},
4316     \texttwosuperior    = {150, 50},
4317     \textthreesuperior  = {150, 50},
4318     \textparagraph      = {100,  },
4319     \textperiodcentered = {500,300},
4320     \textonequarter     = { 50,  },
4321     \textonehalf        = { 50,  },
4322     \textplusminus      = {100,100},
4323     \textmultiply       = {150,150},
4324     \textdivide         = {150,150},
4325 }
4326
4327 </ptm>

```

14.3.3 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).

```

4328 \SetProtrusion
4329 <m-t> [ name = OT1-sc,
4330 <bch> [ name = bch-sc,
4331 <cmr> [ name = cmr-sc-OT1,
4332 <pad> [ name = pad-sc,
4333 <pmn> [ name = pmnj-sc,
4334 <ppl> [ name = ppl-sc,
4335 <ptm> [ name = ptm-sc,
4336 <m-t> load = default ]
4337 <bch> load = bch-default ]
4338 <cmr> load = cmr-OT1 ]
4339 <pad> load = pad-default ]
4340 <pmn> load = pmnj-default ]
4341 <ppl> load = ppl-default ]
4342 <ptm> load = ptm-default ]
4343 <m-t|bch|pad|pmn> { encoding = OT1,
4344 <cmr|ppl|ptm> { encoding = {OT1,OT4},
4345 <bch> family = bch,
4346 <cmr> family = cmr,
4347 <pad> family = {pad,padx,padj},
4348 <pmn> family = pmnj,
4349 <ppl> family = {ppl,pplx,pplj},
4350 <ptm> family = {ptm,ptmx,ptmj},
4351 shape = sc }
4352 {

```



```

4353     a = {50,50},
4354 <cmr|pad|ppl|ptm> \ae = {50, },
4355 <bch|pmn> c = {50, },
4356 <bch|pad|pmn> d = { ,50},
4357 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
4358 <bch|pad|pmn> g = {50, },
4359 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
4360 <bch> j = {100, },
4361 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
4362 <ptm> l = { ,80},
4363 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
4364 <ptm> 013 = { ,80}, % fl
4365 <bch|pad|pmn> o = {50,50},
4366 <bch|pad|pmn> \oe = {50, },
4367 <ppl> p = { 0, 0},
4368 <bch|pad|pmn> q = {50,70},
4369 <ppl> q = { 0, },
4370 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
4371 t = {50,50},
4372 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50},
4373 <ptm> y = {80,80},
4374 }
4375
4376 \SetProtrusion
4377 <m-t> [ name = T1-sc,
4378 <bch> [ name = bch-sc-T1,
4379 <cmr> [ name = cmr-sc-T1,
4380 <pad> [ name = pad-sc-T1,
4381 <pmn> [ name = pmnj-sc-T1,
4382 <ppl> [ name = ppl-sc-T1,
4383 <ptm> [ name = ptm-sc-T1,
4384 <m-t> load = T1-default ]
4385 <bch> load = bch-T1 ]
4386 <cmr> load = cmr-T1 ]
4387 <pad> load = pad-T1 ]
4388 <pmn> load = pmnj-T1 ]
4389 <ppl> load = ppl-T1 ]
4390 <ptm> load = ptm-T1 ]
4391 { encoding = {T1,LV1},
4392 <bch> family = bch,
4393 <cmr> family = cmr,
4394 <pad> family = {pad,padx,padj},
4395 <pmn> family = pmnj,
4396 <ppl> family = {ppl,pplx,pplj},
4397 <ptm> family = {ptm,ptmx,ptmj},
4398 shape = sc }
4399 {
4400     a = {50,50},
4401 <cmr|pad|ppl|ptm> \ae = {50, },
4402 <bch|pmn> c = {50, },
4403 <bch|pad|pmn> d = { ,50},
4404 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
4405 <bch|pad|pmn> g = {50, },
4406 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
4407 <bch> j = {100, },
4408 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
4409 <ptm> l = { ,80},
4410 <m-t|bch|cmr|pad|pmn|ppl> 029 = { ,50}, % fl
4411 <ptm> 029 = { ,80}, % fl
4412 <bch|pad|pmn> o = {50,50},

```

```

4413 <bch|pad|pmn> \oe = {50, },
4414 <ppl> p = { 0, 0},
4415 <bch|pad|pmn> q = {50,70},
4416 <ppl> q = { 0, },
4417 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
4418 t = {50,50},
4419 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50},
4420 <ptm> y = {80,80},
4421 }
4422
4423 <*cmr>
4424 \SetProtrusion
4425 [ name = cmr-sc-T5,
4426 load = cmr-T5 ]
4427 { encoding = T5,
4428 family = cmr,
4429 shape = sc }
4430 {
4431 a = {50,50},
4432 f = { ,50},
4433 j = {50, },
4434 l = { ,50},
4435 r = { , 0},
4436 t = {50,50},
4437 y = {50,50},
4438 }
4439
4440 </cmr>
4441 <*pmn>
4442 \SetProtrusion
4443 [ name = pmnx-sc,
4444 load = pmnj-sc ]
4445 { encoding = OT1,
4446 family = pmnx,
4447 shape = sc }
4448 {
4449 l = {230,180},
4450 }
4451
4452 \SetProtrusion
4453 [ name = pmnx-sc-T1,
4454 load = pmnj-sc-T1 ]
4455 { encoding = {T1,LV1},
4456 family = pmnx,
4457 shape = sc }
4458 {
4459 l = {230,180},
4460 }
4461

```

14.3.4 Italic Small Caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's `fontinstallationguide` suggests `si`.

```

4462 \SetProtrusion
4463 [ name = pmnj-scit,
4464 load = pmnj-it ]
4465 { encoding = OT1,

```

```

4466     family   = pmnj,
4467     shape     = {scit,si} }
4468     {
4469         a = {50,  },
4470     \ae = {  ,-50},
4471         b = {20,-50},
4472         c = {50,-50},
4473         d = {20, 0},
4474         e = {20,-50},
4475         f = {10, 0},
4476     012 = {10,-50}, % fi
4477     013 = {10,-50}, % fl
4478     014 = {10,-50}, % ffi
4479     015 = {10,-50}, % ffi
4480         g = {50,-50},
4481         i = {20,-50},
4482         j = {20, 0},
4483         k = {20,  },
4484         l = {20,50},
4485         m = {  ,-30},
4486         n = {  ,-30},
4487         o = {50,  },
4488     \oe = {50,-50},
4489         p = {20,-50},
4490         q = {50,  },
4491         r = {20, 0},
4492         s = {20,-30},
4493         t = {70,  },
4494         u = {50,-50},
4495         v = {100,  },
4496         w = {100,  },
4497         y = {50,  },
4498         z = {  ,-50},
4499     }
4500
4501 \SetProtrusion
4502 [ name     = pmnj-scit-T1,
4503   load     = pmnj-it-T1 ]
4504 { encoding = {T1,LV1},
4505   family   = pmnj,
4506   shape     = {scit,si} }
4507 {
4508     a = {50,  },
4509 \ae = {  ,-50},
4510     b = {20,-50},
4511     c = {50,-50},
4512     d = {20, 0},
4513     e = {20,-50},
4514     f = {10, 0},
4515     028 = {10,-50}, % fi
4516     029 = {10,-50}, % fl
4517     030 = {10,-50}, % ffi
4518     031 = {10,-50}, % ffi
4519         g = {50,-50},
4520         i = {20,-50},
4521     188 = {20, 0}, % ij
4522         j = {20, 0},
4523         k = {20,  },
4524         l = {20,50},
4525         m = {  ,-30},

```

```

4526     n = { , -30},
4527     o = {50, },
4528     \oe = {50, -50},
4529     p = {20, -50},
4530     q = {50, },
4531     r = {20, 0},
4532     s = {20, -30},
4533     t = {70, },
4534     u = {50, -50},
4535     v = {100, },
4536     w = {100, },
4537     y = {50, },
4538     z = { , -50},
4539 }
4540
4541 \SetProtrusion
4542 [ name      = pmnx-scit,
4543   load      = pmnj-scit ]
4544 { encoding = OT1,
4545   family   = pmnx,
4546   shape     = {scit,si} }
4547 {
4548   1 = {100,150},
4549 }
4550
4551 \SetProtrusion
4552 [ name      = pmnx-scit-T1,
4553   load      = pmnj-scit-T1 ]
4554 { encoding = {T1,LY1},
4555   family   = pmnx,
4556   shape     = {scit,si} }
4557 {
4558   1 = {100,150},
4559 }
4560
4561 </pmn>

```

14.3.5 textcomp

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

```

4562 \SetProtrusion
4563 <m-t> [ name      = textcomp ]
4564 <bch> [ name      = bch-textcomp ]
4565 <cmr> [ name      = cmr-textcomp ]
4566 <pad> [ name      = pad-textcomp ]
4567 <pmn> [ name      = pmn-textcomp ]
4568 <ppl> [ name      = ppl-textcomp ]
4569 <ptm> [ name      = ptm-textcomp ]
4570 <m-t> { encoding = TS1 }
4571 <!m-t> { encoding = TS1,
4572         family   = bch }
4573 <cmr>   family   = cmr }
4574 <pad>   family   = {pad,padx,padj} }
4575 <pmn>   family   = {pmnx,pmnj} }
4576 <ppl>   family   = {ppl,pplx,pplj} }
4577 <ptm>   family   = {ptm,ptmx,ptmj} }
4578 {

```

```

4579 <cmr> \textquotestraightbase = {300,300},
4580 <pad|pmn> \textquotestraightbase = {400,400},
4581 <cmr|pmn> \textquotestraightdblbase = {300,300},
4582 <pad> \textquotestraightdblbase = {400,400},
4583 <bch|cmr|pad|pmn> \texttwelveudash = {200,200},
4584 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
4585 <cmr|pmn> \textquotesingle = {300,400},
4586 <pad> \textquotesingle = {400,500},
4587 <ptm> \textquotesingle = {500,500},
4588 <bch|cmr|pmn> \textasteriskcentered = {200,300},
4589 <pad> \textasteriskcentered = {300,300},
4590 <pmn> \textfractionsolidus = {-200,-200},
4591 <cmr> \textoneoldstyle = {100,100},
4592 <pmn> \textoneoldstyle = { , 50},
4593 <cmr> \textthreeoldstyle = { , 50},
4594 <pad|pmn> \textthreeoldstyle = { 50, },
4595 <cmr> \textfouroldstyle = { 50, 50},
4596 <pad|pmn> \textfouroldstyle = { 50, },
4597 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
4598 <cmr> \textlangle = {400, },
4599 <cmr> \textrangle = { , 400},
4600 <m-t|bch|pmn|ptm> \textminus = {200,200},
4601 <cmr|pad|ppl> \textminus = {300,300},
4602 <bch|pad|pmn> \textlbrackdbl = {100, },
4603 <bch|pad|pmn> \textrbrackdbl = { , 100},
4604 <pmn> \textasciigrave = {200,500},
4605 <bch|cmr|pad|pmn> \texttildelow = {200,250},
4606 <pmn> \textasciibreve = {300,400},
4607 <pmn> \textasciicaron = {300,400},
4608 <pmn> \textacutedbl = {200,300},
4609 <pmn> \textgravedbl = {150,300},
4610 <bch|pmn> \textdagger = { 80, 80},
4611 <cmr|pad> \textdagger = {100,100},
4612 <ptm> \textdagger = {150,150},
4613 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
4614 <ptm> \textdaggerdbl = {100,100},
4615 <bch> \textbardbl = {100,100},
4616 <bch> \textbullet = {200,200},
4617 <cmr|pad|pmn> \textbullet = { , 100},
4618 <ptm> \textbullet = {150,150},
4619 <bch|cmr|pmn> \textcelsius = { 50, },
4620 <pad> \textcelsius = { 80, },
4621 <bch> \textflorin = { 50, 50},
4622 <pad> \textflorin = { , 100},
4623 <pmn> \textflorin = { 50, 100},
4624 <ptm> \textflorin = { 50, 70},
4625 <cmr> \textcolonmonetary = { , 50},
4626 <pad|pmn> \textcolonmonetary = { 50, },
4627 <pmn> \textinterrobang = { , 100},
4628 <pmn> \textinterrobangdown = {100, },
4629 <m-t|pad|ptm> \texttrademark = {100,100},
4630 <bch> \texttrademark = {150,150},
4631 <cmr|ppl> \texttrademark = {200,200},
4632 <pmn> \texttrademark = { 50, 50},
4633 <bch> \textcent = { 50, },
4634 <ptm> \textcent = {100,100},
4635 <bch> \textsterling = { 50, },
4636 <bch> \textbrokenbar = {200,200},
4637 <pmn> \textasciidieresis = {300,400},
4638 <m-t|bch|cmr|pad|ptm> \textcopyright = {100,100},

```

```

4639 <pmn> \textcopyright = {100,150},
4640 <ppl> \textcopyright = {200,200},
4641 <bch|cmr> \textordfeminine = {100,200},
4642 <pad|pmn> \textordfeminine = {200,200},
4643 <bch|cmr|pad|pmn> \textlnot = {200, },
4644 <m-t|bch|cmr|pad|ptm> \textregistered = {100,100},
4645 <pmn> \textregistered = { 50,150},
4646 <ppl> \textregistered = {200,200},
4647 <pmn> \textasciimacron = {150,200},
4648 <m-t|ppl|ptm> \textdegree = {300,300},
4649 <bch> \textdegree = {150,200},
4650 <cmr|pad> \textdegree = {400,400},
4651 <pmn> \textdegree = {150,400},
4652 <bch|cmr|pad|pmn> \textpm = {150,200},
4653 <ptm> \textpm = { 50, 80},
4654 <bch> \texttwosuperior = {100,200},
4655 <cmr> \texttwosuperior = { 50,100},
4656 <pad|pmn> \texttwosuperior = {200,200},
4657 <ptm> \texttwosuperior = { 50, 50},
4658 <bch> \textthreesuperior = {100,200},
4659 <cmr> \textthreesuperior = { 50,100},
4660 <pad|pmn> \textthreesuperior = {200,200},
4661 <ptm> \textthreesuperior = { 50, 50},
4662 <pmn> \textasciicute = {300,400},
4663 <bch> \textmu = { ,100},
4664 <bch|pad|pmn> \textparagraph = { ,100},
4665 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
4666 <ptm> \textperiodcentered = {300,300},
4667 <bch> \textonesuperior = {200,300},
4668 <cmr|pad|pmn> \textonesuperior = {200,200},
4669 <ptm> \textonesuperior = {100,100},
4670 <bch|pad|pmn> \textordmasculine = {200,200},
4671 <cmr> \textordmasculine = {100,200},
4672 <bch|cmr|pmn> \texteuro = {100, },
4673 <pad> \texteuro = { 50,100},
4674 <bch|ptm> \texttimes = {100,100},
4675 <cmr> \texttimes = {150,250},
4676 <pad> \texttimes = {100,150},
4677 <pmn> \texttimes = { 70,100},
4678 <bch|pad|pmn> \textdiv = {150,200},
4679 <cmr> \textdiv = {150,250},
4680 <ptm> \textdiv = { 50,100},
4681 <ptm> \textperthousand = { ,50},

```

All remaining characters can be found in the source.

```

4682 }
4683
4684 <*cmr|pad|pmn>
4685 \SetProtrusion
4686 <cmr> [ name = cmr-textcomp-it ]
4687 <pad> [ name = pad-textcomp-it ]
4688 <pmn> [ name = pmn-textcomp-it ]
4689 { encoding = TS1,
4690 <cmr> family = cmr,
4691 <pad> family = {pad,padx,padj},
4692 <pmn> family = {pmnx,pmnj},
4693 shape = {it,sl} }
4694 {
4695 <cmr> \textquotestraightbase = {300,600},
4696 <pad|pmn> \textquotestraightbase = {400,400},

```

```

4697 <cmr> \textquotestraightdblbase = {300,600},
4698 <pad> \textquotestraightdblbase = {300,400},
4699 <pmn> \textquotestraightdblbase = {300,300},
4700 \texttwelveudash = {200,200},
4701 \textthreequartersemdash = {150,150},
4702 <cmr> \textquotesingle = {600,300},
4703 <pad> \textquotesingle = {800,100},
4704 <pmn> \textquotesingle = {300,200},
4705 <cmr> \textasteriskcentered = {300,200},
4706 <pad> \textasteriskcentered = {500,100},
4707 <pmn> \textasteriskcentered = {200,300},
4708 <pmn> \textfractionsolidus = {-200,-200},
4709 <cmr> \textoneoldstyle = {100, 50},
4710 <pad> \textoneoldstyle = {100, },
4711 <pmn> \textoneoldstyle = { 50, },
4712 <pad> \texttwooldstyle = { 50, },
4713 <pmn> \texttwooldstyle = {-50, },
4714 <cmr> \textthreeoldstyle = {100, 50},
4715 <pmn> \textthreeoldstyle = {-100, },
4716 <cmr> \textfouroldstyle = { 50, 50},
4717 <pad> \textfouroldstyle = { 50,100},
4718 <cmr> \textsevenoldstyle = { 50, 80},
4719 <pad> \textsevenoldstyle = { 50, },
4720 <pmn> \textsevenoldstyle = { 20, },
4721 <cmr> \textlangle = {400, },
4722 <cmr> \textrangle = { ,400},
4723 <cmr|pad> \textminus = {300,300},
4724 <pmn> \textminus = {200,200},
4725 <pad|pmn> \textlbrackdbl = {100, },
4726 <pad|pmn> \textrbrackdbl = { ,100},
4727 <pmn> \textasciigrave = {300,300},
4728 \texttildelow = {200,250},
4729 <pmn> \textasciibreve = {300,300},
4730 <pmn> \textasciicaron = {300,300},
4731 <pmn> \textacutedbl = {200,300},
4732 <pmn> \textgravedbl = {150,300},
4733 <cmr> \textdagger = {100,100},
4734 <pad> \textdagger = {200,100},
4735 <pmn> \textdagger = { 80, 50},
4736 <cmr|pad> \textdaggerdbl = { 80, 80},
4737 <pmn> \textdaggerdbl = { 80, 50},
4738 <cmr> \textbullet = {200,100},
4739 <pad> \textbullet = {300, },
4740 <pmn> \textbullet = { 30, 70},
4741 <cmr> \textcelsius = {100, },
4742 <pad> \textcelsius = {200, },
4743 <pmn> \textcelsius = { 50,-50},
4744 <pad> \textflorin = {100, },
4745 <pmn> \textflorin = { 50,100},
4746 <cmr> \textcolonmonetary = {150, },
4747 <pad> \textcolonmonetary = {100, },
4748 <pmn> \textcolonmonetary = { 50,-50},
4749 <cmr|pad> \texttrademark = {200, },
4750 <pmn> \texttrademark = { 50,100},
4751 <pmn> \textasciidieresis = {300,200},
4752 <cmr> \textcopyright = {100, },
4753 <pad> \textcopyright = {200,100},
4754 <pmn> \textcopyright = {100,150},
4755 <cmr> \textordfeminine = {100,100},
4756 <pmn> \textordfeminine = {200,200},

```

```

4757 <cmr|pad> \textlnot = {300, },
4758 <pmn> \textlnot = {200, },
4759 <cmr> \textregistered = {100, },
4760 <pad> \textregistered = {200,100},
4761 <pmn> \textregistered = { 50,150},
4762 <pmn> \textasciimacron = {150,200},
4763 <cmr|pad> \textdegree = {500,100},
4764 <pmn> \textdegree = {150,150},
4765 <cmr> \textpm = {150,100},
4766 <pad> \textpm = {200,150},
4767 <pmn> \textpm = {150,200},
4768 <cmr> \textonesuperior = {400, },
4769 <pad> \textonesuperior = {300,100},
4770 <pmn> \textonesuperior = {200,100},
4771 <cmr> \texttwosuperior = {400, },
4772 <pad> \texttwosuperior = {300, },
4773 <pmn> \texttwosuperior = {200,100},
4774 <cmr> \textthreesuperior = {400, },
4775 <pad> \textthreesuperior = {300, },
4776 <pmn> \textthreesuperior = {200,100},
4777 <pmn> \textasciicute = {300,200},
4778 <cmr> \textparagraph = {200, },
4779 <pmn> \textparagraph = { ,100},
4780 <cmr> \textperiodcentered = {500,500},
4781 <pad|pmn> \textperiodcentered = {300,400},
4782 <cmr> \textordmasculine = {100,100},
4783 <pmn> \textordmasculine = {200,200},
4784 <cmr> \texteuro = {200, },
4785 <pad> \texteuro = {100, },
4786 <pmn> \texteuro = {100,-50},
4787 <cmr> \texttimes = {200,200},
4788 <pad> \texttimes = {200,100},
4789 <pmn> \texttimes = { 70,100},
4790 <cmr|pad> \textdiv = {200,200},
4791 <pmn> \textdiv = {150,200},
4792 }
4793
4794 </cmr|pad|pmn>

```

14.3.6 Math

Now to the math symbols for Computer Modern Roman. Definitions have been extracted from fontmath.ltx. I did not spend too much time fiddling with these settings, so they can surely be improved.

The math font ‘operators’ (also used for the `\mathrm` and `\mathbf` alphabets) is OT1/cmr, which we’ve already set up above. It’s declared as:

```

\DeclareSymbolFont{operators} {OT1}{cmr}{m}{n}
\SetSymbolFont{operators}{bold}{OT1}{cmr}{bx}{n}

```

`\mathit` (OT1/cmr/m/it) is also already set up.

There are (for the moment) no settings for `\mathsf` and `\mathtt`.

Math font ‘letters’ (also used as `\mathnormal`) is declared as:

```

\DeclareSymbolFont{letters} {OML}{cmm}{m}{it}
\SetSymbolFont{letters} {bold}{OML}{cmm}{b}{it}

```



```

4795 <*cmr>
4796 \SetProtrusion
4797 [ name      = cmr-math-letters ]
4798 { encoding = OML,
4799   family   = cmm,
4800   series   = {m,b},
4801   shape    = it }
4802 {
4803   A = {100, 50}, % \mathnormal
4804   B = { 50,   },
4805   C = { 50,   },
4806   D = { 50, 50},
4807   E = { 50,   },
4808   F = {100, 50},
4809   G = { 50, 50},
4810   H = { 50, 50},
4811   I = { 50, 50},
4812   J = {150, 50},
4813   K = { 50,100},
4814   L = { 50, 50},
4815   M = { 50,   },
4816   N = { 50,   },
4817   O = { 50,   },
4818   P = { 50,   },
4819   Q = { 50, 50},
4820   R = { 50,   },
4821   S = { 50,   },
4822   T = { 50,100},
4823   U = { 50, 50},
4824   V = {100,100},
4825   W = { 50,100},
4826   X = { 50,100},
4827   Y = {100,100},
4828   f = {100,100},
4829   h = {   ,100},
4830   i = {   , 50},
4831   j = {   , 50},
4832   k = {   , 50},
4833   r = {   , 50},
4834   v = {   , 50},
4835   w = {   , 50},
4836   x = {   , 50},
4837   "0B = { 50,100}, % \alpha
4838   "0C = { 50, 50}, % \beta
4839   "0D = {200,150}, % \gamma
4840   "0E = { 50, 50}, % \delta
4841   "0F = { 50, 50}, % \epsilon
4842   "10 = { 50,150}, % \zeta
4843   "11 = {   ,   }, % \eta
4844   "12 = { 50,   }, % \theta
4845   "13 = {   ,100}, % \iota
4846   "14 = {   ,100}, % \kappa
4847   "15 = {100, 50}, % \lambda
4848   "16 = {   , 50}, % \mu
4849   "17 = {   , 50}, % \nu
4850   "18 = {   , 50}, % \xi
4851   "19 = { 50,100}, % \pi
4852   "1A = { 50, 50}, % \rho
4853   "1B = {   ,150}, % \sigma
4854   "1C = { 50,150}, % \tau

```

```

4855 "1D = { 50, 50}, % \upsilon
4856 % "1E = { , }, % \phi
4857 "1F = { 50,100}, % \chi
4858 "20 = { 50, 50}, % \psi
4859 "21 = { , 50}, % \omega
4860 "22 = { , 50}, % \varepsilon
4861 "23 = { , 50}, % \vartheta
4862 "24 = { , 50}, % \varpi
4863 "25 = {100, }, % \varrho
4864 "26 = {100,100}, % \varsigma
4865 "27 = { 50, 50}, % \varphi
4866 "28 = {100,100}, % \leftharpoonup
4867 "29 = {100,100}, % \leftharpoondown
4868 "2A = {100,100}, % \rightharpoonup
4869 "2B = {100,100}, % \rightharpoondown
4870 "2C = {300,200}, % \lhook
4871 "2D = {200,300}, % \rhook
4872 "2E = { ,100}, % \triangleright
4873 "2F = {100, }, % \triangleleft
4874 % 0 - 9
4875 "3A = { ,500}, % ., \ldotp
4876 "3B = { ,500}, % ,
4877 "3C = {200,100}, % <
4878 "3D = {300,400}, % /
4879 "3E = {100,200}, % >
4880 "3F = {200,200}, % \star
4881 % "40 = { , }, % \partial
4882 "5B = { ,100}, % \flat
4883 % "5C = { , }, % \natural
4884 % "5D = { , }, % \sharp
4885 "5E = {200,200}, % \smile
4886 "5F = {200,200}, % \frown
4887 % "60 = { , }, % \ell
4888 % "7B = { , }, % \imath
4889 "7C = {100, }, % \jmath
4890 "7D = { ,100}, % \wp
4891 }
4892

```

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}

```

```

4893 \SetProtrusion
4894 [ name = cmr-math-symbols ]
4895 { encoding = OMS,
4896   family = cmsy,
4897   series = {m,b},
4898   shape = n }
4899 {
4900   A = {150, 50}, % \mathcal
4901   C = { ,100},
4902   D = { , 50},
4903   F = { 50,150},
4904   I = { ,100},
4905   J = {100,150},
4906   K = { ,100},
4907   L = {100, },
4908   M = { 50, 50},
4909   N = { 50,100},

```

```

4910     P = {    , 50},
4911     Q = { 50,   },
4912     R = {    , 50},
4913     T = { 50,150},
4914     V = { 50, 50},
4915     W = {    , 50},
4916     X = {100,100},
4917     Y = {100,   },
4918     Z = {100,150},
4919     "00 = {300,300}, % -
4920     "01 = {    ,700}, % \cdot, \cdotp
4921     "02 = {150,250}, % \times
4922     "03 = {150,250}, % *, \ast
4923     "04 = {200,300}, % \div
4924     "05 = {150,250}, % \diamond
4925     "06 = {200,200}, % \pm
4926     "07 = {200,200}, % \mp
4927     "08 = {100,100}, % \oplus
4928     "09 = {100,100}, % \ominus
4929     "0A = {100,100}, % \otimes
4930     "0B = {100,100}, % \oslash
4931     "0C = {100,100}, % \odot
4932     "0D = {100,100}, % \bigcirc
4933     "0E = {100,100}, % \circ
4934     "0F = {100,100}, % \bullet
4935     "10 = {100,100}, % \asymp
4936     "11 = {100,100}, % \equiv
4937     "12 = {200,100}, % \subseteq
4938     "13 = {100,200}, % \supseteq
4939     "14 = {200,100}, % \leq
4940     "15 = {100,200}, % \geq
4941     "16 = {200,100}, % \preceq
4942     "17 = {100,200}, % \succeq
4943     "18 = {200,200}, % \sim
4944     "19 = {150,150}, % \approx
4945     "1A = {200,100}, % \subset
4946     "1B = {100,200}, % \supset
4947     "1C = {200,100}, % \ll
4948     "1D = {100,200}, % \gg
4949     "1E = {300,100}, % \prec
4950     "1F = {100,300}, % \succ
4951     "20 = {100,200}, % \leftarrow
4952     "21 = {200,100}, % \rightarrow
4953     "22 = {100,100}, % \uparrow
4954     "23 = {100,100}, % \downarrow
4955     "24 = {100,100}, % \leftrightarrows
4956     "25 = {100,100}, % \nearrow
4957     "26 = {100,100}, % \searrow
4958     "27 = {100,100}, % \simeq
4959     "28 = {100,100}, % \Leftarrow
4960     "29 = {100,100}, % \Rightarrow
4961     "2A = {100,100}, % \Uparrow
4962     "2B = {100,100}, % \Downarrow
4963     "2C = {100,100}, % \Leftrightarrow
4964     "2D = {100,100}, % \nrightarrow
4965     "2E = {100,100}, % \swarrow
4966     "2F = {    ,100}, % \propto
4967     "30 = {    ,400}, % \prime
4968     "31 = {100,100}, % \infty
4969     "32 = {150,100}, % \in

```

```

4970 "33 = {100,150}, % \ni
4971 "34 = {100,100}, % \triangle, \bigtriangleup
4972 "35 = {100,100}, % \bigtriangledown
4973 % "36 = { , }, % \not
4974 % "37 = { , }, % \mapstochar
4975 "38 = { ,100}, % \forall
4976 "39 = {100, }, % \exists
4977 "3A = {200, }, % \neg
4978 % "3B = { , }, % \emptyset
4979 % "3C = { , }, % \Re
4980 % "3D = { , }, % \Im
4981 "3E = {200,200}, % \top
4982 "3F = {200,200}, % \bot, \perp
4983 % "40 = { , }, % \aleph
4984 % "5B = { , }, % \cup
4985 % "5C = { , }, % \cap
4986 % "5D = { , }, % \uplus
4987 "5E = {100,200}, % \wedge
4988 "5F = {100,200}, % \vee
4989 "60 = { ,300}, % \vdash
4990 "61 = {300, }, % \dashv
4991 "62 = {100,100}, % \lfloor
4992 "63 = {100,100}, % \rfloor
4993 "64 = {100,100}, % \lceil
4994 "65 = {100,100}, % \rceil
4995 "66 = {150, }, % \lbrace
4996 "67 = { ,150}, % \rbrace
4997 "68 = {400, }, % \langle
4998 "69 = { ,400}, % \rangle
4999 % "6A = { , }, % \arrowvert, \mid, \vert, |
5000 % "6B = { , }, % \Arrowvert, \parallel, \Vert
5001 "6C = {100,100}, % \updownarrow
5002 "6D = {100,100}, % \Updownarrow
5003 "6E = {100,300}, % \, \backslash, \setminus
5004 % "6F = { , }, % \wr
5005 % "70 = { , }, % \sqrt{\text{sign}}
5006 % "71 = { , }, % \amalg
5007 "72 = {100,100}, % \nabla
5008 % "73 = { , }, % \smallint
5009 % "74 = { , }, % \sqcup
5010 % "75 = { , }, % \sqcap
5011 % "76 = { , }, % \sqsubseteq
5012 % "77 = { , }, % \sqsupseteq
5013 % "78 = { , }, % \mathsection
5014 "79 = {200,200}, % \dagger
5015 "7A = {100,100}, % \ddagger
5016 "7B = {100, }, % \mathparagraph
5017 "7C = {100,100}, % \clubsuit
5018 "7D = {100,100}, % \diamondsuit
5019 "7E = {100,100}, % \heartsuit
5020 "7F = {100,100}, % \spadesuit
5021 }
5022

```

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}
```

```
5023 %\SetProtrusion
```

```

5024 % [ name      = cmr-math-largesymbols ]
5025 % { encoding = OMX,
5026 %   family    = {cmex,cmr} }
5027 % {
5028 %   "00 % (
5029 %   "01 % )
5030 %   "02 % [
5031 %   "03 % ]
5032 %   "04 % \lfloor
5033 %   "05 % \rfloor
5034 %   "06 % \lceil
5035 %   "07 % \rceil
5036 %   "08 % \lbrace
5037 %   "09 % \rbrace
5038 %   "0A % <, \angle
5039 %   "0B % >, \angle
5040 %   "0C % \vert, |
5041 %   "0D % \Vert
5042 %   "0E % /
5043 %   "0F % \backslash
5044 %   "3A % \lggroup
5045 %   "3B % \rgroup
5046 %   "3C % \arrowvert
5047 %   "3D % \Arrowvert
5048 %   "3E % \bracevert
5049 %   "3F % \updownarrow
5050 %   "40 % \lmoustache
5051 %   "41 % \rmoustache
5052 %   "46 % \bigsqcup
5053 %   "48 % \oint
5054 %   "4A % \bigodot
5055 %   "4C % \bigoplus
5056 %   "4E % \bigotimes
5057 %   "50 % \sum
5058 %   "51 % \prod
5059 %   "52 % \intop
5060 %   "53 % \bigcup
5061 %   "54 % \bigcap
5062 %   "55 % \biguplus
5063 %   "56 % \bigwedge
5064 %   "57 % \bigvee
5065 %   "60 % \coprod
5066 %   "70 % \sqrt{\hspace{0.5em}}
5067 %   "77 % \Updownarrow
5068 %   "78 % \uparrow
5069 %   "79 % \downarrow
5070 %   "7A % \braced, \lmoustache
5071 %   "7B % \bracerd, \rmoustache
5072 %   "7C % \bracelu
5073 %   "7D % \braceru
5074 %   "7E % \Uparrow
5075 %   "7F % \Downarrow
5076 % }
5077 </cmr>
5078 <!/cfg-u>

```

14.3.7 AMS fonts

```

5079 <*/cfg-u>

```

Symbol font ‘a’, defined in amssymb.

```
\DeclareSymbolFont{AMSA}{U}{msa}{m}{n}
```

```
5080 *msa
5081 \SetProtrusion
5082   [ name      = AMSa ]
5083   { encoding = U,
5084     family   = msa }
5085   {
5086 % "00 = { , }, % \boxdot
5087 % "01 = { , }, % \boxplus
5088 % "02 = { , }, % \boxtimes
5089 % "03 = { , }, % \square
5090 % "04 = { , }, % \blacksquare
5091 % "05 = {150,250}, % \centerdot
5092 % "06 = {100,100}, % \lozenge
5093 % "07 = { 50, 50}, % \blacklozenge
5094 % "08 = { 50, 50}, % \circlearrowright
5095 % "09 = { 50, 50}, % \circlearrowleft
5096 % "0A = { 50, 50}, % \rightleftharpoons
5097 % "0B = { 50, 50}, % \leftrightharpoons
5098 % "0C = { , }, % \boxminus
5099 % "0D = {-50,150}, % \Vdash
5100 % "0E = {-50,150}, % \Vvdash
5101 % "0F = {-70,150}, % \vDash
5102 % "10 = {100,100}, % \twoheadrightarrow
5103 % "11 = { 50,150}, % \twoheadleftarrow
5104 % "12 = { 50,100}, % \leftleftarrows
5105 % "13 = { 50, 80}, % \rightrightarrows
5106 % "14 = {120,120}, % \upuparrows
5107 % "15 = {120,120}, % \downdownarrows
5108 % "16 = {200,200}, % \upharpoonright
5109 % "17 = {200,200}, % \downharpoonright
5110 % "18 = {200,200}, % \upharpoonleft
5111 % "19 = {200,200}, % \downharpoonleft
5112 % "1A = { 50, 80}, % \rightarrowtail
5113 % "1B = { 50, 80}, % \leftarrowtail
5114 % "1C = { 50, 50}, % \leftrightarrows
5115 % "1D = { 50, 50}, % \rightleftarrows
5116 % "1E = {150, }, % \Lsh
5117 % "1F = { ,150}, % \Rsh
5118 % "20 = { 50,100}, % \rightsquigarrow
5119 % "21 = { 50, 50}, % \leftrightsquigarrow
5120 % "22 = { 50, 50}, % \looparrowleft
5121 % "23 = { 50, 50}, % \looparrowright
5122 % "24 = { , 50}, % \circeq
5123 % "25 = { ,100}, % \succsim
5124 % "26 = { ,100}, % \gtrsim
5125 % "27 = { ,100}, % \gtrapprox
5126 % "28 = { 50, 50}, % \multimap
5127 % "29 = { , }, % \therefore
5128 % "2A = { , }, % \because
5129 % "2B = { 50, 50}, % \doteqdot
5130 % "2C = { 50, 50}, % \triangleq
5131 % "2D = {100, 50}, % \precsim
5132 % "2E = {100, 50}, % \lesssim
5133 % "2F = { 50, 50}, % \lessapprox
5134 % "30 = {100, 50}, % \eqslantless
5135 % "31 = { 50, 50}, % \eqslantgtr
5136 % "32 = {100, 50}, % \curlyeqprec
```

```

5137 "33 = { 50,100}, % \curlyeqsucc
5138 "34 = {100, 50}, % \preccurlyeq
5139 % "35 = { , }, % \leqq
5140 "36 = { 50, }, % \leqslant
5141 % "37 = { , }, % \lessgtr
5142 "38 = { , 50}, % \backprime
5143 "39 = {200,200}, % \dabar@ : the dash bar in \dash(left,right)arrow
5144 % "3A = { , }, % \risingdotseq
5145 % "3B = { , }, % \fallingdotseq
5146 "3C = { 50,100}, % \succcurlyeq
5147 % "3D = { , }, % \geqq
5148 "3E = { , 50}, % \geqslant
5149 % "3F = { , }, % \gtrless
5150 "40 = { , 50}, % \sqsubset
5151 "41 = { 50, }, % \sqsupset
5152 "42 = { ,150}, % \vartriangleright, \rhd
5153 "43 = {150, }, % \vartriangleleft, \lhd
5154 "44 = { ,100}, % \trianglerighteq, \unrhd
5155 "45 = {100, }, % \trianglelefteq, \unlhd
5156 "46 = {100,100}, % \bigstar
5157 % "47 = { , }, % \between
5158 "48 = { 50, 50}, % \blacktriangledown
5159 "49 = { ,100}, % \blacktriangleright
5160 "4A = {100, }, % \blacktriangleleft
5161 "4B = { ,150}, % \dashrightarrow (the arrow)
5162 "4C = {150, }, % \dashleftarrow
5163 "4D = { 50, 50}, % \vartriangle
5164 "4E = { 50, 50}, % \blacktriangle
5165 "4F = { 50, 50}, % \triangledown
5166 "50 = { 50, 50}, % \eqcirc
5167 % "51 = { , }, % \lesseqgtr
5168 % "52 = { , }, % \gtreqless
5169 % "53 = { , }, % \lesseqgtr
5170 % "54 = { , }, % \gtreqless
5171 % "55 = { , }, % \yen
5172 "56 = { ,150}, % \Rrightarrow
5173 "57 = {150, }, % \Lleftarrow
5174 "58 = {100,200}, % \checkmark
5175 % "59 = { , }, % \veebar
5176 % "5A = { , }, % \barwedge
5177 % "5B = { , }, % \doublebarwedge
5178 "5C = { 50, 50}, % \angle
5179 "5D = { 50, 50}, % \measuredangle
5180 "5E = { 50, 50}, % \sphericalangle
5181 "5F = { , 50}, % \varpropto
5182 "60 = {100,100}, % \smallsmile
5183 "61 = {100,100}, % \smallfrown
5184 "62 = { 50, }, % \Subset
5185 "63 = { , 50}, % \Supset
5186 % "64 = { , }, % \Cup
5187 % "65 = { , }, % \Cap
5188 "66 = {100,100}, % \curlywedge
5189 "67 = {100,100}, % \curlyvee
5190 "68 = { 50,100}, % \leftthreetimes
5191 "69 = {100, 50}, % \rightthreetimes
5192 % "6A = { , }, % \subteqq
5193 % "6B = { , }, % \supseteqq
5194 "6C = { 50, 50}, % \bumpeq
5195 "6D = { 50, 50}, % \Bumpeq
5196 "6E = {100, }, % \lll

```

```

5197 "6F = { ,100}, % \ggg
5198 "70 = { 50,100}, % \ulcorner
5199 "71 = {100, 50}, % \urcorner
5200 % "72 = { , }, % \circledR
5201 % "73 = { , }, % \circledS
5202 % "74 = { , }, % \pitchfork
5203 "75 = {100,150}, % \dotplus
5204 "76 = { 50,100}, % \backsim
5205 % "77 = { , }, % \backsimeq
5206 "78 = { 50,100}, % \llcorner
5207 "79 = {100, 50}, % \lrcorner
5208 % "7A = { , }, % \maltese
5209 % "7B = { , }, % \complement
5210 "7C = {100,100}, % \intercal
5211 "7D = { 50, 50}, % \circledcirc
5212 "7E = { 50, 50}, % \circledast
5213 "7F = { 50, 50}, % \circleddash
5214 }
5215
5216 </msa>

```

Symbol font ‘b’.

```

\DeclareSymbolFont{AMSb}{U}{msb}{m}{n}
\DeclareSymbolFontAlphabet{\mathbb}{AMSb}

```

```

5217 <*msb>
5218 \SetProtrusion
5219 [ name = AMSb ]
5220 { encoding = U,
5221   family = msb }
5222 {
5223   A = { 50, 50}, % \mathbb
5224   C = { 50, 50},
5225   G = { , 50},
5226   L = { , 50},
5227   P = { , 50},
5228   R = { , 50},
5229   T = { , 50},
5230   V = { 50, 50},
5231   X = { 50, 50},
5232   Y = { 50, 50},
5233   "00 = { 50, 50}, % \lvertneqq
5234   "01 = { 50, 50}, % \gvertneqq
5235   "02 = { 50, 50}, % \lneq
5236   "03 = { 50, 50}, % \ngeq
5237   "04 = {100, 50}, % \lless
5238   "05 = { 50,150}, % \ngtr
5239   "06 = {100, 50}, % \nprec
5240   "07 = { 50,150}, % \nsucc
5241   "08 = { 50, 50}, % \lneqq
5242   "09 = { 50, 50}, % \gneqq
5243   "0A = {100,100}, % \lneqslant
5244   "0B = {100,100}, % \ngeqslant
5245   "0C = {100, 50}, % \lneq
5246   "0D = { 50,100}, % \gneq
5247   "0E = {100, 50}, % \npreceq
5248   "0F = { 50,100}, % \nsucceq
5249   "10 = { 50, }, % \precnsim
5250   "11 = { 50, 50}, % \succnsim
5251   "12 = { 50, 50}, % \lsim

```



```

5252 "13 = { 50, 50}, % \gnsim
5253 "14 = { 50, 50}, % \nleqq
5254 "15 = { 50, 50}, % \ngeqq
5255 "16 = { 50, 50}, % \precneqq
5256 "17 = { 50, 50}, % \succneqq
5257 "18 = { 50, 50}, % \precnapprox
5258 "19 = { 50, 50}, % \succnapprox
5259 "1A = { 50, 50}, % \lnapprox
5260 "1B = { 50, 50}, % \gnapprox
5261 "1C = {150,200}, % \nsim
5262 "1D = { 50, 50}, % \ncong
5263 "1E = {100,150}, % \diagup
5264 "1F = {100,150}, % \diagdown
5265 "20 = {100, 50}, % \varsubsetneq
5266 "21 = { 50,100}, % \varsupsetneq
5267 "22 = {100, 50}, % \nsubseteqq
5268 "23 = { 50,100}, % \supseteqq
5269 "24 = {100, 50}, % \subsetneqq
5270 "25 = { 50,100}, % \supsetneqq
5271 "26 = {100, 50}, % \varsubsetneqq
5272 "27 = { 50,100}, % \varsupsetneqq
5273 "28 = {100, 50}, % \subsetneq
5274 "29 = { 50,100}, % \supsetneq
5275 "2A = {100, 50}, % \nsubseteq
5276 "2B = { 50,100}, % \nsupseteq
5277 "2C = { 50,100}, % \nparallel
5278 "2D = {100,150}, % \nmid
5279 "2E = {150,150}, % \nshortmid
5280 "2F = {100,100}, % \nshortparallel
5281 "30 = { ,150}, % \nvDash
5282 "31 = { ,150}, % \nVDash
5283 "32 = { ,100}, % \nvDash
5284 "33 = { ,100}, % \nVDash
5285 "34 = { ,100}, % \ntrianglerighteq
5286 "35 = {100, }, % \ntrianglelefteq
5287 "36 = {100, }, % \ntriangleleft
5288 "37 = { ,100}, % \ntriangleright
5289 "38 = {100,200}, % \leftarrow
5290 "39 = {100,200}, % \rightarrow
5291 "3A = {100,100}, % \Leftarrow
5292 "3B = { 50,100}, % \Rightarrow
5293 "3C = {100,100}, % \Leftrightarrow
5294 "3D = {100,200}, % \leftrightharrow
5295 "3E = { 50, 50}, % \divideontimes
5296 "3F = { 50, 50}, % \varnothing
5297 % "40 = { , }, % \nexists
5298 "60 = {200, }, % \Finv
5299 "61 = { , 50}, % \Game
5300 % "66 = { , }, % \mho
5301 % "67 = { , }, % \eth
5302 "68 = {100,100}, % \eqsim
5303 "69 = { 50, }, % \beth
5304 "6A = { 50, }, % \gimel
5305 "6B = {150, }, % \daleth
5306 "6C = {200, }, % \lessdot
5307 "6D = { ,200}, % \gtrdot
5308 "6E = {100,200}, % \ltimes
5309 "6F = {150,100}, % \rtimes
5310 "70 = { 50,100}, % \shortmid
5311 "71 = { 50, 50}, % \shortparallel

```

```

5312 "72 = {200,300}, % \smallsetminus
5313 "73 = {100,200}, % \thicksim
5314 "74 = { 50,100}, % \thickapprox
5315 "75 = { 50, 50}, % \approx
5316 "76 = { 50,100}, % \succapprox
5317 "77 = { 50, 50}, % \precapprox
5318 "78 = {100,100}, % \curvearrowleft
5319 "79 = { 50,150}, % \curvearrowright
5320 "7A = { 50,200}, % \digamma
5321 "7B = {100, 50}, % \varkappa
5322 % "7C = { , }, % \Bbbk
5323 % "7D = { , }, % \hslash
5324 % "7E = { , }, % \hbar
5325 "7F = {200, }, % \backepsilon
5326 }
5327
5328 </msb>

```

Euler Fraktur font (eufrak).

```

\DeclareMathAlphabet{\mathfrak}{U}{euf}{m}{n}
\SetMathAlphabet{\mathfrak}{bold}{U}{euf}{b}{n}

```

```

5329 <*euf>
5330 \SetProtrusion
5331 [ name = mathfrak ]
5332 { encoding = U,
5333   family = euf }
5334 {
5335   A = { , 50},
5336   B = { , 50},
5337   C = { 50, 50},
5338   D = { , 80},
5339   E = { 50, },
5340   G = { , 50},
5341   L = { , 80},
5342   O = { , 50},
5343   T = { , 80},
5344   X = { 80, 50},
5345   Z = { 80, 50},
5346   b = { , 50},
5347   c = { , 50},
5348   k = { , 50},
5349   p = { , 50},
5350   q = { 50, },
5351   v = { , 50},
5352   w = { , 50},
5353   x = { , 50},
5354   1 = {100,100},
5355   2 = { 80, 80},
5356   3 = { 80, 50},
5357   4 = { 80, 50},
5358   7 = { 50, 50},
5359 }
5360
5361 </euf>

```

Euler script font (euca1).

```

\DeclareMathAlphabet\EuScript{U}{eus}{m}{n}
\SetMathAlphabet\EuScript{bold}{U}{eus}{b}{n}

```

```

5362 <*eus>
5363 \SetProtrusion
5364   [ name      = euscript ]
5365   { encoding = U,
5366     family   = eus   }
5367   {
5368     A = {100,100},
5369     B = { 50,100},
5370     C = { 50, 50},
5371     D = { 50,100},
5372     E = { 50,100},
5373     F = { 50,  },
5374     G = { 50,  },
5375     H = {  ,100},
5376     K = {  , 50},
5377     L = {  ,150},
5378     M = {  , 50},
5379     N = {  , 50},
5380     O = { 50, 50},
5381     P = { 50, 50},
5382     T = {  ,100},
5383     U = {  , 50},
5384     V = { 50, 50},
5385     W = { 50, 50},
5386     X = { 50, 50},
5387     Z = { 50,100},
5388   }
5389
5390 </eus>
5391 </cfg-u>
5392 <*beta>

```

14.4 Interword Spacing

Default unit is space.

```

5393 %%% -----
5394 %%% INTERWORD SPACING SETTINGS
5395
5396 \SetExtraSpacing
5397   [ name = default ]
5398   { encoding = {OT1,T1,LY1,OT4,T5} }
5399   {

```

These settings are only a first approximation. The following reasoning is from a mail from Ulrich Dirr. I do not claim to have coped with the task.

‘The idea is – analog to the tables for expansion and protrusion – to have tables for optical reduction/expansion of spaces in dependence of the actual character so that the distance between words is optically equal.

When reducing distances the (weighting) order is:

- after commas

```

5400           { , } = { , -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)

5401 r = { , -300, 300 },

- before or after lowercase characters with ascenders

5402 b = { , -200, 200 },

5403 d = { , -200, 200 },

5404 f = { , -200, 200 },

5405 h = { , -200, 200 },

5406 k = { , -200, 200 },

5407 l = { , -200, 200 },

5408 t = { , -200, 200 },

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

5409 c = { , -100, 100 },

5410 p = { , -100, 100 },

5411 v = { , -100, 100 },

5412 w = { , -100, 100 },

5413 z = { , -100, 100 },

5414 x = { , -100, 100 },

5415 y = { , -100, 100 }, % ?

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

5416 i = { , 50, -50 },

5417 m = { , 50, -50 },

5418 n = { , 50, -50 },

5419 u = { , 50, -50 },

- after colon and semicolon

5420 : = { , 200, -200 },

5421 ; = { , 200, -200 },

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

5422 . = { , 250, -250 },

5423 ! = { , 250, -250 },

5424 ? = { , 250, -250 },

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

5425 }

5426

Questions are:

- Is the result really better?
- Is it overdone? (Try with a factor < 1000.)
- Should the first parameter also be used? (Probably.)

The following settings simulate \nonfrenchspacing (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the \TeX book:

‘If the space factor f is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [...] Then the stretch component is multiplied by $f / 1000$, while the shrink component is multiplied by $1000 / f$.’

The ‘extra space’ (\fontdimen7) for Computer Modern Roman is a third of \fontdimen2 , i. e., 333.

```
5427 \SetExtraSpacing
5428   [ name      = nonfrench-cmr,
5429     load       = default,
5430     context    = nonfrench ]
5431   { encoding = {OT1,T1,LY1,OT4,T5},
5432     family    = cmr }
5433   {
```

`latex.ltx` has:

```
\def\nonfrenchspacing{
\sffcode'\. 3000
```

```
5434   . = {333,2000,-667},
```

```
\sffcode'\? 3000
```

```
5435   ? = {333,2000,-667},
```

```
\sffcode'\! 3000
```

```
5436   ! = {333,2000,-667},
```

```
\sffcode'\: 2000
```

```
5437   : = {333,1000,-500},
```

```
\sffcode'\; 1500
```

```
5438   ; = {   , 500,-333},
```

```
\sffcode'\, 1250
```

```
5439   {,}= {   , 250,-200},
```

```
}
```

```
5440 }
```

```
5441
```

`fontinst`, however, which is also used to create the PSNFSS font metrics, sets it to 240 by default. Therefore, the fallback settings use this value for the first component.

```
5442 \SetExtraSpacing
5443   [ name      = nonfrench-default,
5444     load       = default,
5445     context    = nonfrench ]
```

```

5446 { encoding = {OT1,T1,LY1,OT4,T5} }
5447 {
5448     . = {240,2000,-667},
5449     ? = {240,2000,-667},
5450     ! = {240,2000,-667},
5451     : = {240,1000,-500},
5452     ; = {    , 500,-333},
5453     {,}= {    , 250,-200},
5454 }
5455

```

14.5 Additional Kerning

Default unit is 1em.

```

5456 %%% -----
5457 %%% ADDITIONAL KERNING
5458

```

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

```

5459 \SetExtraKerning
5460 [ name = empty ]
5461 { encoding = {OT1,T1,LY1,OT4,T5,TS1} }
5462 { }
5463
5464 \SetExtraKerning
5465 [ name      = french-default,
5466   context   = french,
5467   unit      = space ]
5468 { encoding = {OT1,T1,LY1} }
5469 {
5470     : = {1000,}, % = \fontdimen2
5471     ; = {500, }, % ~ \thinspace
5472     ! = {500, },
5473     ? = {500, },
5474 }
5475

```

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

```

5476 \SetExtraKerning
5477 [ name      = french-guillemets,
5478   context   = french-guillemets,
5479   load      = french-default,
5480   unit      = space ]
5481 { encoding = {OT1,T1,LY1} }
5482 {
5483     \guillemotleft = { ,800}, % = 0.8\fontdimen2
5484     \guillemotright = {800, },
5485 }
5486
5487 \SetExtraKerning
5488 [ name      = turkish,
5489   context   = turkish,
5490   unit      = space ]
5491 { encoding = {OT1,T1,LY1} }
5492 {
5493     : = {500, }, % ~ \thinspace
5494     ! = {500, },

```

```

5495     {=} = {500, },
5496   }
5497

```

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

```

5498%% The settings for the commands \lsstyle and \textls.
5499\SetExtraKerning
5500  [ name      = letterspacing-default,
5501    context   = letterspacing,
5502    unit      = 1em,
5503    preset    = {1000,1000} ]
5504  { encoding  = {OT1,OT4} }
5505  {

```

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

```

5506    . = {0, },
5507    0 = {0,0},
5508    1 = {0,0},
5509    2 = {0,0},
5510    3 = {0,0},
5511    4 = {0,0},
5512    5 = {0,0},
5513    6 = {0,0},
5514    7 = {0,0},
5515    8 = {0,0},
5516    9 = {0,0},
5517    \textquoteleft  = {0,0},    \textquoteright  = {0,0},
5518    \textquotedblleft = {0,0},    \textquotedblright = {0,0},
5519  }
5520
5521\SetExtraKerning
5522  [ name      = letterspacing-T1,
5523    load      = letterspacing-default,
5524    context   = letterspacing,
5525    unit      = 1em,
5526    preset    = {1000,1000} ]
5527  { encoding  = {T1,LY1,T5} }
5528  {
5529    \quotesinglbase = {0,0},    \quotedblbase   = {0,0},
5530    \guilsinglleft  = {0,0},    \guilsinglright = {0,0},
5531    \guillemotleft  = {0,0},    \guillemotright = {0,0},
5532  }
5533
5534</beta>
5535</config>

```

15 Auxiliary File for Micro Fine Tuning

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

```

5536< *test>
5537\documentclass{article}
5538
5539%% Here you can set the font you want to test, using
5540%% the commands \fontfamily, \fontseries, and \fontshape.
5541%% Make sure to end all lines with a comment character!

```

```

5542 \newcommand*{\TestFont}{%
5543   \fontfamily{ppl}%
5544   %% \fontseries{b}%
5545   %% \fontshape{it}% sc, sl
5546 }
5547
5548 \usepackage{ifthen}
5549 \usepackage[T1]{fontenc}
5550 %%\usepackage[latin1]{inputenc}
5551 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
5552
5553 \pagestyle{empty}
5554 \setlength{\parindent}{0pt}
5555 \newcommand*{\crulefill}{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
5556 \newcommand*{\testprotrusion}[2][{}]{%
5557   \ifthenelse{\equal{#1}{r}}{\{#2}%
5558     lorem ipsum dolor sit amet,
5559     \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
5560     \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
5561     you know the rest%
5562   \ifthenelse{\equal{#1}{l}}{\{#2}%
5563     \linebreak
5564     {\fontencoding{\encodingdefault}%
5565      \fontseries{\seriesdefault}%
5566      \fontshape{\shapedefault}%
5567      \selectfont
5568      Here is the beginning of a line, \dotfill and here is its end}\linebreak
5569 }
5570 \newcommand*{\showTestFont}{\expandafter\stripprefix\meaning\TestFont}
5571 \def\stripprefix#1>{}
5572 \newcount\charcount
5573 \begin{document}
5574
5575 \microtypesetup{expansion=false}
5576
5577 {\centering The font in this document is called by:\\
5578 \texttt{\showTestFont}\par}\bigskip
5579
5580 \TestFont\selectfont
5581 This line intentionally left empty\linebreak
5582 %% A -- Z
5583 \charcount=65
5584 \loop
5585   \testprotrusion{\char\charcount}
5586   \advance\charcount 1
5587   \ifnum\charcount < 91 \repeat
5588 %% a -- z
5589 \charcount=97
5590 \loop
5591   \testprotrusion{\char\charcount}
5592   \advance\charcount 1
5593   \ifnum\charcount < 123 \repeat
5594 %% 0 -- 9
5595 \charcount=48
5596 \loop
5597   \testprotrusion{\char\charcount}
5598   \advance\charcount 1
5599   \ifnum\charcount < 58 \repeat
5600 %%
5601 \testprotrusion[r]{,}

```



```

5602 \testprotrusion[r]{.}
5603 \testprotrusion[r]{;}
5604 \testprotrusion[r]{:}
5605 \testprotrusion[r]{?}
5606 \testprotrusion[r]{!}
5607 \testprotrusion[l]{\textexclamdown}
5608 \testprotrusion[l]{\textquestiondown}
5609 \testprotrusion[r]{}}
5610 \testprotrusion[l]{(}
5611 \testprotrusion{/}
5612 \testprotrusion{\char'\}
5613 \testprotrusion{-}
5614 \testprotrusion{\textendash}
5615 \testprotrusion{\textemdash}
5616 \testprotrusion{\textquoteleft}
5617 \testprotrusion{\textquoteright}
5618 \testprotrusion{\textquotedblleft}
5619 \testprotrusion{\textquotedblright}
5620 \testprotrusion{\quotesinglbase}
5621 \testprotrusion{\quotedblbase}
5622 \testprotrusion{\guilsinglleft}
5623 \testprotrusion{\guilsinglright}
5624 \testprotrusion{\guillemotleft}
5625 \testprotrusion{\guillemotright}
5626
5627 \bigskip
5628 The following displays the current font stretched by 5%,
5629 normal, and shrunk by 5\%:
5630
5631 \microtypesetup{expansion=true}
5632
5633 \bigskip
5634 \newlength{\MTln}
5635 \newcommand*{\teststring}
5636 {ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
5637 \settowidth{\MTln}{\teststring}
5638
5639 \parbox{1.05\MTln}{\teststring\linebreak\}
5640 \parbox{0.95\MTln}{\teststring\par\bigskip}
5641 \parbox{0.95\MTln}{\teststring}
5642 \end{document}
5643 /test

```

Needless to say that things may always be improved. For suggestions, mail to w.m.l@gmx.net.

A Change History

Version 1.0 (2004/09/11)

General: Initial version 1

Version 1.1 (2004/09/21)

General: configuration file names in lowercase (suggested by Harald Harders)	54	family has already been loaded	55
issue an error instead of a warning, when pdfTeX version is too old for autoexpand	99	\MT@get@basefamily: only remove suffix, if it is 'x' or 'j'	56
remove 8-bit characters from the configuration files (suggested by Harald Harders)	102	\MT@get@listname@: don't check for empty characteristics list	56
Protrusion: add factors for some more characters	107	\MT@ifempty: bug fix: use category code 12 for the percent character (reported by Tom Kink)	32
settings for Adobe Minion (contributed by Harald Harders)	107	\MT@is@number: numbers may also be specified in hexadecimal or octal (suggested by Harald Harders)	60
\DeclareCharacterInheritance: new macro: possibility to specify character inheritance	82	\MT@pdftex@no: bug fix concerning version check (reported by Harald Harders)	25
\MT@declare@sets: remove spaces around set name	66	\MT@permute: don't use sets for empty encoding	84
\MT@DeclareSet@: remove spaces around first argument	66	\MT@pr@split: bug fix: allow zero and negative values	41
\MT@find@file: bug fix: also check whether the file for the base font		\MT@use@set: remove spaces around set name	71
		\UseMicrotypeSet: remove spaces around first argument	71

Version 1.2 (2004/10/03)

Font Sets: declare cmr as an alias for cmor	74	\MT@get@highlevel: check whether defaults have changed	67
new: allmath and basicmath	73	\MT@get@listname@: alternatively check for alias font name	56
General: check for packages that might load fonts	63	\MT@get@size: additional magic to catch some errors	69
Protrusion: add settings for Adobe Garamond and Computer Modern Roman in TS1 encoding	124	hijack \set@fontsize instead of \setfontsize	69
add settings for Computer Modern Roman math symbols	128	\MT@get@slot: bug fix: group must also include \MT@get@composite	58
\MT@check@default: new macro	68	\MT@loop: bug fix: new macro, used instead of \loop	35
\MT@context: bug fix: set inheritance list \globally to \empty	58	\MT@maybe@do: also check for alias font name	37
\MT@define@inh@key@encoding: check whether only one encoding specified	83	\MT@permute@@@@: more sanity checks for \SetProtrusion and \SetExpansion	85
\MT@familyalias: define alias font name as an alternative, not as a replacement	37	\MT@setupfont: also search for alias font file	37
\MT@get@basefamily: also remove 'w' (swash capitals)	56	bug fix: call \@@enc@update if necessary	37

Version 1.3 (2004/10/27)

Font Sets: declare cmr as an alias for aer, zer and hfor	74	\MT@get@codes@name: bug fix: specifying load option does no longer require to give a name, too	77
\MT@catcodes: check some category codes (compatibility with german)	26	\MT@load@list: check whether list exists	54

Version 1.4 (2004/11/12)

General: don't use scratch registers in global definitions	58	\microtypesetup: bug fix: set the correct levels, and remember them; warning when enabling an option disabled in package options	87
no need to check for packages that might load fonts anymore	63	\MT@pdfcprot@error: check for pdfcprot	28
use \pickup@font instead of \define@newfont as the hook for \MT@setupfont	63	\MT@set@ex@opt: bug fix: specifying extra options does no longer require to give a name, too	77
use one instead of five counters	35		
Protrusion: tweak quote characters for cmr variants (OT1, T1, lmr)	111		

Version 1.4a (2004/11/17)

General: new option: final	93	ing files (reported by Michael Hoppe)	55
\MT@begin@catcodes: bug fix: reset some more catcodes when read-			

Version 1.4b (2004/11/26)

General: bug fix: set catcodes before reading global configuration file (reported by Christoph Bier)	95	\MT@get@basefamily: bug fix: failed for font names of the form abczz (reported by Georg Verwey)	56
new message if \pdfoutput is changed	96	\MT@get@slot: don't define \MT@char globally (save stack problem)	58
optimization: use less \csnames and \expandafters	30	\MT@ifdimen: don't set \MT@count globally (save stack problem)	32
Protrusion: harmonize dashes in upshape and italic (cmr, pad, ppl)	107	\MT@in@clist: bug fix: compare with \ instead of \relax (discovered by Herb Schulz)	34
slanted like italics	113	\MT@use@set: don't use undeclared font sets	71
\MT@checklist@family: bug fix: don't try alias family name if encoding failed	39		

Version 1.5 (2004/12/15)

Documentation: add note about DVIOutput option	7	Hàn Thế Thành)	93
add short history (section 12)	20	defaults: calculate step as min(stretch,shrink)/5	98
General: defaults: step: 4 (suggested by		defaults: turn off expansion for DVI	

output	96	babel)	55
disable automatic expansion for DVI		\MT@catcodes: reset catcode of “~” (com-	
output	98	patibility with chemsym)	26
new option: selected, by default		\MT@get@highlevel: don’t test defaults	
false (suggested by Hàn Thế		if called after begin document	67
Thành)	92	\MT@scale@factor: warning for factors	
Inheritance: remove \ss from T1 list,		outside limits	43
add \DJ	103	\MT@scale@to@em: don’t use \lcode and	
Protrusion: settings for Bitstream Char-		\rcode for the calculation	41
ter	107	\MT@set@ex@codes: allow non-selected	
\DeclareMicrotypeAlias: remove		font expansion	46
spaces around arguments	72	\MT@set@pr@codes: adjust protrusion	
\MT@begin@catcodes: reset catcode of		factors before setting the inher-	
“=” (compatibility with Turkish		iting characters	40

Version 1.6 (2005/01/24)

General: defaults: turn off expansion for		tune CMR math letters (OML encod-	
old pdf \TeX versions	94	ing)	128
disable automatic expansion for old		\MT@def@num@opt: test whether numeric	
pdf \TeX versions	99	options receive a number	93
load a font, if none is active	30	\MT@get@charwd: use e- \TeX ’s	
new option: factor, by default 1000	93	\fontcharwd, if available	42
restructure dtx file	102	\MT@get@inh@list: correct message if	
test whether \pickup@font has		selected is false	58
changed	65	\MT@set@ex@codes: introduce factor op-	
use e- \TeX ’s \ifcsname and		tion	46
\ifdefined if defined	31	\MT@set@pr@codes: introduce factor op-	
Protrusion: add italic uppercase Greek		tion	40
letters	113	\MT@use@set: retain current set if new	
improve settings for numbers		set is undeclared	71
(pointed out by Peter Muthesius)	108	\MT@vinfo: new macro: used instead of	
		\ifMT@verbose	24

Version 1.6a (2005/02/02)

Documentation: add table of fonts with		ported by Bernard Gaulle)	58
tailored protrusion settings	15	\MT@pdftex@no: new macro	25
\MT@get@slot: completely redone, hope-		\MT@reset@ef@codes: only reset	
fully more robust (compatible		\efcodes for older pdf \TeX ver-	
with frenchpro; problem re-		sions	46

Version 1.7 (2005/03/23)

Documentation: add hint about compat-		matic expansion for old pdf \TeX	
ibility	18	versions	81
remove table of match order	12	allow specification of size ranges	
General: \SetExpansion: bug fix: re-		(suggested by Andreas Böhmann)	68
move space after autoexpand	81	modify \showhyphens	99
\SetExpansion: don’t allow auto-		new value for verbose option:	

errors	93	<code>\MT@if@list@exists</code> : don't define	
shorter command names	36	<code>\MT@#1@c@name</code> \globally, here	
warning when running in draft		and elsewhere	57
mode	97	<code>\MT@increment</code> : use e-TeX's <code>\numexpr</code> if	
Protrusion: fix: remove \ from OT1, add		available	35
<code>\textbackslash</code> to T1 encoding	109	<code>\MT@is@composite</code> : new macro: con-	
<code>\DeclareMicrotypeAlias</code> : may also be		struct command for composite	
used inside configuration files ..	72	character; no uncontrolled expan-	
<code>\LoadMicrotypeFile</code> : new macro (sug-		sion	62
gested by Andreas Böhmann) ...	72	<code>\MT@scale</code> : new macro: use e-TeX's	
<code>\Microtype@Hook</code> : new macro for font		<code>\numexpr</code> if available	35
package authors	95	<code>\MT@set@ex@codes</code> : two versions of this	
<code>\microtypesetup</code> : bug fix: warn-		macro	46
ing also when setting to		<code>\MT@split@name</code> : don't define	
(no)compatibility	87	<code>\MT@encoding &c.</code> \globally	37
<code>\MT@begin@catcodes</code> : also use inside		<code>\MT@test@ast</code> : make it simpler	68
configuration commands	55	<code>\MT@try@order</code> : always check for size,	
reset catcode of ':' (compatibility		too (suggested by Andreas Böh-	
with french* packages)	55	mann)	56
<code>\MT@get@listname@</code> : use <code>\@tfor</code> (An-		bug fix: also check for	
dreas Böhmann's idea)	56	<code>//<series>/<shape>/</code> (reported by	
<code>\MT@get@size</code> : comparison with 1 to al-		Andreas Böhmann)	56
low size smaller than 1 (suggested		<code>\MT@warn@code@too@large</code> : new macro:	
by Andreas Böhmann)	69	type out maximum protrusion fac-	
<code>\MT@get@slot</code> : remove backslash hack	58	tor	43
test for <code>\chardef</code> commands ...	59	<code>\MT@warn@err</code> : new macro: for	
test whether <code>\<encoding>\<.></code> is de-		verbose=errors	24
fined	59		

Version 1.8 (2005/06/23)

Documentation: add example for factor		Times	112
option	12	settings for AMS math fonts	107
add example of how to get rid of a		verified settings for slanted Com-	
widow	13	puter Modern Roman	118
add hint about error messages ...	19	<code>\DeclareMicrotypeAlias</code> : warning	
Font Sets: add U encoding to <code>allmath</code> ..	73	when overriding an alias font ...	72
declare <code>ppl</code> and <code>ptm</code> as alias fonts for		<code>\DeclareMicrotypeSetDefault</code> : new	
<code>pxr</code> resp. <code>txr</code>	74	macro: set fall back font set	71
General: <code>\SetProtrusion</code> : new option:		<code>\MT@add@accent</code> : bug fix: disable	
unit	80	micro-typographic setup in-	
if font substitution has occurred, set		side <code>\add@accent</code> (reported by	
up the substitute font, not the		Stephan Hennig)	65
selected one	64	<code>\MT@begin@catcodes</code> : reset catcodes of	
new option: config to load a differ-		the remaining ASCII characters	55
ent main configuration file	94	<code>\MT@check@rlist</code> : made recursive	86
new option: unit, by default		<code>\MT@curr@list@name</code> : new macro: cur-	
character	94	rent list type and name	62
Inheritance: remove <code>\DJ</code> from T1 list		<code>\MT@declare@sets</code> : warning when re-	
(it's the same as <code>\DH</code>)	103	defining a set	66
Protrusion: add LY1 characters for		<code>\MT@define@set@key@</code> : use comma lists	

instead of token lists	67	\MT@map@clist@n: new macro: used in-	
\MT@find@file: no longer wrap names		stead of \@for	33
in commands	55	\MT@map@tlist@n: new macro: used in-	
\MT@get@charwd: warning for missing		stead of \@tfor	34
(resp. zero-width) characters . . .	42	\MT@old@cmd: renamed commands	
\MT@get@dimen@six: new macro: test		from \..MicroType.. to	
whether \fontdimen6 is defined	40	\..MicroType..	24
\MT@get@listname@: made recursive . .	56	\MT@pdftex@no: case 5: pdf \TeX 1.30 . .	25
\MT@get@slot: bug fix: expand active		\MT@permute@@@@: add ranges to the	
characters	59	beginning of the lists	85
test whether \<encoding>\<.> is de-		\MT@pr@splitt: get character width once	
defined made more robust	59	only	41
\MT@get@unit: new macro: get unit for		\MT@scale: bug fix: remove spaces in	
codes	44	non-e- \TeX variant (reported by	
\MT@in@list: made recursive	34	Mark Rossi)	35
\MT@is@active: new macro: translate		\MT@setupfont@hook: restore \% and \#	
inputenc-defined characters . . .	60	when hyperref is loaded	29
\MT@is@letter: warning for non-ASCII		restore csquotes's active characters	29
characters	60	restore percent character if Spanish	
\MT@led@kern: character protrusion with		babel is loaded	29
ledmac	28	\MT@use@set: bug fix: remove braces in	
\MT@make@string: use		first line	71
\@onelevel@sanitize	36	\MT@xadd: simplified	33

Version 1.9 (2005/10/28)

Documentation: add hint about		option unit: rename value relative	
verbatim environment	18	to character	94
add remark about Type 1 fonts re-		warning if user requested zero step	98
quired for automatic font expan-		Inheritance: add list for OT4	104
sion	6	add list for T5 (requested by Hàn	
Font Sets: add OT4 encoding to text sets	73	Thế Thành)	105
add T5 encoding to text sets	73	Protrusion: fix: remove uppercase Greek	
declare qpl and qtm (qfonts) as alias		letters from T1 encoded CMR . .	110
fonts for ppl resp. ptm	74	settings for OT4 encoding (Com-	
General: \DeclareMicrotypeSet: new		puter Modern Roman, Palatino,	
key: font	70	Times)	107
\SetProtrusion and \SetExpansion:		settings for T5 encoding (Computer	
new key: font	79	Modern Roman)	107
\SetProtrusion: value 'relative' re-		\DisableLigatures: new macro: disable	
named to 'character' for option		ligatures (requires pdf \TeX 1.30) . .	73
unit	80	\microtypecontext: new macro: change	
allow context-specific font setup . .	64	setup context in the document . .	89
disable \DeclareMicrotypeSetDefault		\MT@checklist@family: bug fix: add two	
after configuration file has been		missing \expandafters	39
loaded	95	\MT@define@option: fix: use true as the	
disable expansion if both step and		default value	91
shrink are zero	99	\MT@detokenize: fix the non-e- \TeX ver-	
disable microtype setup inside		sion	62
hyperref's \pdfstringdef (re-		\MT@exp@two@n: new macros: less	
ported by Hàn Thế Thành)	30	\expandafters	31

<code>\MT@get@opt</code> : new key ‘preset’ to set all characters to the specified value before loading the lists	44	<code>\MT@maybe@do</code> : redone	37
<code>\MT@is@active</code> : redone: use <code>\set@display@protect</code>	60	<code>\MT@pdf@tex@no</code> : compatibility with \TeX Live hack (reported by Herbert Voß)	25
<code>\MT@is@letter</code> : using <code>\catcode</code> should be more efficient than inspecting the <code>\meaning</code>	60	<code>\MT@rem@from@list</code> : new macro: remove an item from a comma list	34
		<code>\MT@scale@factor</code> : generalized	43
		<code>\mt@toks</code> : no longer use <code>\toks@</code>	27

Version 2.0 ()

General: <code>\SetExtraKerning/\SetExtraSpacing</code> : fault 100	93	<code>\DeclareMicrotypeBabelHook</code> : new command: interaction with babel	73
new key ‘space’ for option unit	80	<code>\MT@pdf@tex@no</code> : case 6: pdf \TeX 1.3x	25
new option: babel, by default false (language-dependant setup suggested by Ulrich Dirr)	92	<code>\textls</code> : new command: letterspacing	53
new option: letterspacing, by de-			

B Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Options:	<code>\microtypecontext</code>	17
DVIoutput	<code>\microtypesetup</code>	8
activate	<code>\SetExpansion</code>	13
auto	<code>\SetProtrusion</code>	11
config	<code>\UseMicrotypeSet</code>	10
draft	<code>\@@enc@update</code>	547
expansion		
factor	A	
final	a0poster (package)	69
protrusion	activate (option)	5, 92
selected	<code>\add@accent</code>	1712, 1713
shrink	ae (package)	16, 74
step	ams* (package)	69
stretch	amssymb (package)	134
unit	auto (option)	6, 92
verbose		
	B	
User Commands:	babel (option)	92, 151
<code>\DeclareCharacterInheritance</code>	babel (package)	2, 22, 29, 55, 73, 74, 101, 102, 148, 150, 151
<code>\DeclareMicrotypeAlias</code>		
<code>\DeclareMicrotypeSet</code>	C	
<code>\DeclareMicrotypeSet*</code>	<code>\cf@encoding</code>	547
<code>\DeclareMicrotypeSetDefault</code>	chemsym (package)	29, 148
<code>\DisableLigatures</code>	CJK (package)	18, 65
<code>\LoadMicrotypeFile</code>		

- cm-super (package) 6
color (package) 7, 96
config (option) 8, 15, 21, 94, 149
contour (package) 96
crop (package) 96
csquotes (package) . 20, 21, 29, 110, 150
\curr@fontshape 1270, 1275, 1674
- D**
- \DeclareCharacterInheritance
..... 9, 15, 2471, 3164
\DeclareMicrotypeAlias
..... 5, 16, 41, 2022, 3165
\DeclareMicrotypeBabelHook . 16, 2068
\DeclareMicrotypeSet
..... 2, 8, 20, 26, 35, 1734, 3151
\DeclareMicrotypeSet* 8, 1734
\DeclareMicrotypeSetDefault
..... 4, 10, 1992, 3095, 3097
\define@newfont 1676
\DisableLigatures 8, 17, 28, 2054
\do@subst@correction 1673
docstrip (package) 23
draft (option) 7, 92, 93
dsfont (package) 40
DVIoutput (option) ... 7, 21, 92, 96, 147
- E**
- eco (package) 16, 74
\efcode 951, 987, 988, 1041, 1044
eucal (package) 138
eufrak (package) 138
expansion (option) 5, 10, 91, 92, 96
- F**
- \f@encoding 547
\f@size 1270, 1275, 1674
factor (option) .. 6, 12, 21, 93, 94, 148
fancyvrb (package) 19
final (option) 7, 22, 93, 147
fix-cm (package) 14
\font 265
\font@name 1676, 1688, 1690, 1692
\fontcharwd 755
\fontdimen 691, 698, 769, 871, 874
fontenc (package) 18
fontinst (package) 141
fontinstallationguide (package) .. 122
french (package) 149
frenchpro (package) 21, 61, 148
- G**
- german (package) 22, 26, 147
- graphics (package) 7, 96
- H**
- hfoldsty (package) 16, 74
hyperref (package) 7, 29, 30, 64, 96, 150
- I**
- IEEEtran (package) 64
\ifcat 1536, 1544
\iffontchar 781
\ifMT@auto 140, 3224, 3270
\ifMT@babel 140, 3357
\ifMT@do 566, 642, 1320
\ifMT@draft 140, 3140
\ifMT@DVIoutput 140, 3127
\ifMT@expansion 140, 3188, 3268
\ifMT@inlist@ 424, 476,
593, 610, 626, 635, 654, 1272,
1351, 1361, 1449, 1697, 2040, 2837
\ifMT@kerning
..... 140, 1266, 3323, 3348, 3381
\ifMT@noligatures 140
\ifMT@nonselected 930, 1461, 1470
\ifMT@norest 1524, 1531
\ifMT@opt@auto 2885, 3227, 3239
\ifMT@opt@expansion 2885, 3133
\ifMT@protrusion 140, 192, 3172
\ifMT@selected 140, 3254, 3273
\ifMT@spacing 140, 3299, 3340
ifpdf (package) 7, 96
inputenc (package) 18, 59–62, 150
- J**
- jurabib (package) 64
- K**
- keyval (package) 27, 36, 41, 67
\knaccode . 1210, 1211, 1223, 1227, 1235
\knbccode
73, 1203, 1204, 1222, 1226, 1234
\knbscode . 1091, 1092, 1117, 1122, 1131
- L**
- ledmac (package) ... 20, 21, 28, 64, 150
ledpar (package) 20
\leftmarginkern 201
letterspacing (option) 93, 151
lineno (package) 28
listings (package) 19
lmodern (package) 6, 112
\LoadMicrotypeFile
..... 10, 16, 44, 2036, 3166
\lpcode 683, 722, 723, 880, 884
\lststyle 18, 259, 1264, 1302

M

- mathpazo (package) 18
- memoir (package) 64
- \Microtype@Hook 95, 3101
- \microtypecontext 12, 17, 1286, 2829, 3168, 3307, 3364, 3367, 3375, 3378
- \microtypesetup 8, 11, 2729, 3167
- minimal (package) 30
- \MT@abbr@ex 518
- \MT@abbr@ex@c 518
- \MT@abbr@ex@inh 518
- \MT@abbr@kn 518
- \MT@abbr@kn@c 518
- \MT@abbr@kn@inh 518
- \MT@abbr@nl 518, 1309
- \MT@abbr@pr 518
- \MT@abbr@pr@c 518
- \MT@abbr@pr@inh 518
- \MT@abbr@sp 518
- \MT@abbr@sp@c 518
- \MT@abbr@sp@inh 518
- \MT@add@accent 1712
- \MT@auto 939, 1037, 3223, 3236, 3247, 3251
- \MT@auto@ 927, 939, 943, 1006
- \MT@autofalse 139, 3234, 3246
- \MT@autotruer 139, 3065
- \MT@babelfalse 139
- \MT@babeltrue 139
- \MT@begin@catcodes 1352, 1353, 1374, 2044, 2165, 2189, 2211, 2232, 2473, 3078, 3079
- \MT@catcodes 115, 136
- \MT@char 712, 722, 723, 728, 729, 732, 734, 749, 755, 781, 880, 881, 883–885, 975, 987, 988, 990, 992, 1041, 1043, 1044, 1080, 1091, 1092, 1098, 1099, 1105, 1106, 1109, 1111, 1117–1119, 1121–1124, 1192, 1203, 1204, 1210, 1211, 1214, 1216, 1222, 1223, 1225–1227, 1502, 1507, 1517, 1520, 1521, 1526, 2553, 2554, 2559, 2562
- \Mt@char 1502, 1509, 1516, 1520, 1537, 1539, 1545, 1546, 1548, 1561, 1562, 1565, 1566, 1569, 1570, 1574, 1576, 1606, 1651
- \MT@charstring 1517, 1605, 1610
- \MT@check@default 1809, 1822
- \MT@check@range 2691, 2693
- \MT@check@range@ 2694, 2696
- \MT@check@rlist 2632, 2682
- \MT@check@rlist@ 2683, 2685
- \MT@checklist@ 573, 586, 1314
- \MT@checklist@family 621
- \MT@checklist@font 605
- \MT@checklist@size 649
- \MT@checksetup 2739, 2750, 2757, 2778, 2800
- \MT@clist@break 392, 599, 612, 643, 660
- \MT@clist@function 392
- \MT@cnt@encoding 2574, 2581, 2582
- \MT@cnt@family 2579, 2589, 2590
- \MT@cnt@series 2587, 2597, 2598
- \MT@cnt@shape 2595, 2604, 2605
- \MT@config@file 3068, 3076, 3077, 3081, 3085
- \MT@context ... 1425, 1437, 1456, 1481
- \MT@count 341, 342, 488, 741, 750, 751, 755, 756, 761, 762, 766, 769, 805, 865
- \MT@curr@contexts 1692, 1696, 1699, 1706, 2831, 2835, 2836, 2838, 2866
- \MT@curr@list@name 674, 808, 924, 1002, 1065, 1177, 1338, 1494, 1649, 1654, 1659, 1665, 1670
- \MT@curr@set@name 1766, 1768–1770, 1772, 1773, 1785, 1787, 1793, 1795, 1804, 1806, 1807, 1811, 1841, 1845, 1876, 1907, 1917, 1920, 2256, 2258
- \MT@declare@char@inh 2478, 2479, 2481, 2482, 2488, 2494
- \MT@declare@codes 2355, 2388, 2389, 2391, 2392
- \MT@declare@sets 1751, 1752, 1754, 1755, 1761, 1767, 2057
- \MT@DeclareSet 22, 23, 25, 1738, 1739, 1741, 3153, 3154, 3156
- \MT@DeclareSet@ 1742, 1745, 1748
- \MT@DeclareSetAndUseIt 1736, 1737, 1741
- \MT@def@bool@opt 2961, 2979
- \MT@def@n 274, 2072, 2169, 2193, 2215, 2236, 2358, 2445, 2498
- \MT@def@num@opt 3015, 3030
- \MT@default@ex@set .. 2010, 3090, 3277
- \MT@default@kn@set .. 2010, 3093, 3330
- \MT@default@pr@set .. 2010, 3089, 3181
- \MT@default@sp@set .. 2010, 3092, 3316
- \MT@define@code@key 2299, 2371–2374, 2531–2533
- \MT@define@code@key@font 2323, 2376, 2535

- \MT@define@code@key@size 2310, 2375, 2534
 \MT@define@context
 ... 2848, 2860, 2861, 2863, 2864
 \MT@define@ex@c@key . 2441, 2454–2456
 \MT@define@inh@key@encoding 2514, 2530
 \MT@define@inh@keys 2529
 \MT@define@key@unit . 2413, 2438, 2439
 \MT@define@option ... 2887, 2923, 2924
 \MT@define@option@ .. 2930, 2958, 2959
 \MT@define@optionX .. 2729, 2788, 2789
 \MT@define@optionX@ . 2791, 2821, 2822
 \MT@define@set@key@ . 1777–1780, 1784
 \MT@define@set@key@font ... 1782, 1911
 \MT@define@set@key@size ... 1781, 1834
 \MT@define@set@keys 1776,
 1954, 1955, 1957, 1958, 1960
 \MT@detokenize 1511, 1643
 \MT@dimen@six 690, 742, 766, 805
 \MT@doc@contexts
 ... 1700, 2836, 2838, 2839, 2866
 \MT@dofalse
 565, 577, 598, 615, 631, 659, 1318
 \MT@dotrue
 565, 569, 595, 628, 637, 656, 1310
 \MT@draftfalse 139, 2981, 2992
 \MT@drafttrue 139, 2987
 \MT@DVIOutputfalse 139
 \MT@DVIOutputtrue 139
 \MT@edef@n 275, 1460,
 1485, 1806, 1981, 1985, 2014,
 2019, 2030, 2251, 2305, 2328,
 2509, 2519, 2654, 2662, 2854,
 2907, 2909, 2944, 2946, 3021, 3036
 \MT@encoding 556, 1420, 1432,
 1511, 1601, 1636, 1658, 1664, 1670
 \MT@encoding@check 2517, 2522
 \MT@end@catcodes . 1354, 1392, 2051,
 2170, 2194, 2216, 2237, 2492, 3080
 \MT@error 45, 183, 1720
 \MT@etex@no
 266, 287, 489, 500, 747, 771, 1643
 \MT@ex@c@name 923–925,
 1007, 1008, 1013, 1014, 1019,
 1020, 1025, 1026, 1031, 1032,
 1048, 1462, 2172, 2193, 2270, 2274
 \MT@ex@context 2832, 2866
 \MT@ex@do 926, 965
 \MT@ex@factor 152, 940, 1011
 \MT@ex@factor@
 ... 940, 955, 960, 977, 978, 1006
 \MT@ex@inh@name 989, 990, 992
 \MT@ex@level 152, 3269, 3271
 \MT@ex@max 167, 980, 981
 \MT@ex@min 167, 983, 984
 \MT@ex@setname
 ... 1978, 3274, 3275, 3277, 3278
 \MT@ex@split 967, 971
 \MT@exp@one@n . 280, 544, 546, 1360,
 1695, 1707, 1761, 1808, 1974,
 2005, 2039, 2488, 2836, 3364, 3375
 \MT@exp@string
 ... 279, 541, 693, 775, 782, 1474
 \MT@exp@two@c 281,
 297, 543, 1510, 1517, 1604, 1644
 \MT@exp@two@n 282, 608, 624, 633, 1269
 \MT@expansion 549, 917, 3294
 \MT@expansionfalse
 139, 3134, 3145, 3264
 \MT@expansiontrue 139, 3064
 \MT@extra@auto 2172, 2461, 2467
 \MT@extra@context
 2150, 2172, 2197, 2218,
 2384, 2629, 2635, 2639, 2643,
 2646, 2650, 2651, 2654, 2690, 2866
 \MT@extra@factor . 2150, 2172, 2197,
 2218, 2267–2269, 2272, 2274, 2368
 \MT@extra@preset
 ... 2150, 2172, 2197, 2218, 2379
 \MT@extra@shrink 2172
 \MT@extra@step 2172
 \MT@extra@stretch 2172
 \MT@extra@unit 2150, 2197,
 2218, 2395, 2405, 2416, 2425, 2428
 \MT@factor@default
 177, 3033, 3039, 3175
 \MT@family
 . 544, 556, 625, 1421, 2031, 2032
 \MT@familyalias . 545, 546, 562, 632,
 634, 636, 638, 1431, 1433, 2033
 \MT@feat .. 566, 695, 786, 788, 790,
 791, 794, 795, 801, 803, 806,
 813–817, 819, 821–825, 828,
 829, 834, 838, 841, 844, 847,
 848, 857, 858, 863, 1328, 1330,
 1333, 1334, 1338–1340, 1342,
 1457, 1458, 1460, 1464, 1465,
 1469, 1473, 1476, 1483, 1485,
 1487, 1488, 1490, 1493, 1496, 1499
 \MT@file@list 1348, 1350,
 1357, 1360, 1364, 1367, 2039, 2043
 \MT@find@file 544, 546, 1348
 \MT@font 541, 543, 609, 611, 614, 683,
 684, 691, 693, 698, 722, 723,

- 728, 729, 749, 755, 769, 775,
781, 782, 871, 874, 880, 881,
884, 885, 927, 943, 951, 987,
988, 1041, 1044, 1091, 1092,
1098, 1099, 1105, 1106, 1117–
1119, 1122–1124, 1131–1133,
1203, 1204, 1210, 1211, 1222,
1223, 1226, 1227, 1234, 1235,
1321, 1404, 1474, 1672, 1687,
1688, 1690, 1695, 1699, 1703, 1708
- \MT@get@axis
... 1927–1930, 1943, 2340–2343
- \MT@get@basefamily 1359, 1393
- \MT@get@char@unit
..... 713, 836, 861, 1081, 1193
- \MT@get@charwd 747, 766, 839, 861
- \MT@get@codes@name
... 2159, 2183, 2205, 2226, 2240
- \MT@get@config 3068
- \MT@get@dimen@six 670, 690, 1061, 1173
- \MT@get@ex@opt 920, 934, 1006
- \MT@get@font 1915, 1923
- \MT@get@font@and@size 2327, 2336
- \MT@get@font@dimen 768, 842
- \MT@get@highlevel
... 1790, 1798, 1945, 2304, 2518
- \MT@get@inh@list
..... 672, 922, 1063, 1175, 1481
- \MT@get@inh@name 2496, 2503
- \MT@get@listname 1403, 1458, 1483
- \MT@get@listname@ 1403
- \MT@get@opt 671, 812, 1062, 1174
- \MT@get@range 1838, 1848, 2314
- \MT@get@size 1855,
1860, 1871, 1890, 1938, 2352
- \MT@get@slot 711,
974, 1079, 1191, 1502, 2552, 2558
- \MT@get@space@unit . 836, 913, 1089,
1096, 1103, 1155, 1159, 1201, 1208
- \MT@get@unit 844, 851
- \MT@get@unit@ 851
- \MT@gobble@settings 690
- \MT@gobblethree 2875, 2879
- \MT@hop@else@fi ... 285, 287, 771, 1305
- \MT@hop@fi 285, 304, 778, 1307
- \MT@if@list@exists
... 669, 919, 933, 1060, 1172, 1456
- \MT@ifdefined@c
..... 194, 222, 239, 287, 731,
989, 1108, 1213, 1283, 1459,
1484, 2031, 2257, 2267, 2618,
2628, 3089, 3090, 3092, 3093,
3101, 3105, 3178, 3274, 3313, 3327
- \MT@ifdefined@n
..... 287, 562, 571, 588, 607,
623, 651, 732, 813, 821, 847,
990, 1007, 1013, 1019, 1025,
1031, 1109, 1214, 1312, 1333,
1427, 1439, 1511, 1769, 1980,
1984, 2012, 2026, 2241, 2242,
2248, 2295, 2506, 2582, 2590,
2598, 2605, 2629, 2643, 2661,
2677, 2906, 2943, 3111, 3362, 3373
- \MT@ifdimen 339, 1902, 2404, 2427, 3053
- \MT@ifempty 321,
720, 726, 1088, 1095, 1102,
1200, 1207, 1297, 1750, 1760,
1800, 1818, 1849, 1850, 1863,
1864, 1924, 1931, 1946, 1963,
1973, 1994, 2004, 2337, 2345,
2357, 2362, 2367, 2378, 2383,
2443, 2477, 2487, 2523, 2736,
2797, 2855, 2892, 2935, 2963,
2982, 2998, 3017, 3032, 3043, 3070
- \MT@ifeq 348,
465, 466, 1882, 2697–2699, 2709
- \MT@ifgt .. 348, 1873, 2701, 2702, 2715
- \MT@iflt 348, 470, 471, 2710, 2711, 2714
- \MT@ifnumber 332, 1568, 2444, 3020, 3035
- \MT@ifstreq 369, 1255, 1329,
1398–1400, 1533, 1706, 2397,
2403, 2418, 2424, 2426, 2738,
2745, 2749, 2756, 2799, 2805,
2895, 2897, 2900, 2903, 2938,
2940, 2966, 2967, 2985, 2986,
3001, 3002, 3007, 3046, 3052, 3303
- \MT@in@clist ... 424, 590, 608, 624,
633, 1269, 1350, 1360, 1695, 2039
- \MT@in@rlist 457, 652, 1448
- \MT@in@rlist@ 457
- \MT@in@rlist@e 457
- \MT@in@tlist 445, 2836
- \MT@in@tlist@ 448, 450
- \MT@increment
..... 488, 2581, 2589, 2597, 2604
- \MT@info
45, 59, 541, 1982, 2741, 2747,
2753, 2760, 2802, 2807, 2878, 2882
- \Mt@info 1610
- \MT@info@nl 45, 56,
60, 195, 540, 2997, 3077, 3129,
3132, 3138, 3174, 3179, 3182,
3186, 3270, 3275, 3278, 3295,

- 3312, 3314, 3317, 3321, 3326,
3328, 3331, 3334, 3344, 3352, 3358
- \MT@inh@do 1495, 2543
- \MT@inh@name 1493, 2561, 2568
- \MT@inh@split 2545, 2549
- \MT@inlist@false ... 423, 426, 446, 458
- \MT@inlist@true 423, 429, 453, 467, 472
- \MT@is@active 1510, 1581
- \MT@is@char 1517, 1604, 1610
- \MT@is@composite 1514, 1632
- \MT@is@letter . 1508, 1535, 1607, 1640
- \MT@is@number 1551, 1556
- \MT@is@symbol 1512, 1598
- \MT@iterate 481
- \MT@kerning 552, 1165, 3355
- \MT@kerningfalse 139, 3148, 3354
- \MT@kerningtrue 139, 1279
- \MT@kn@c@name 1176–
1178, 1243, 1249, 2218, 2236
- \MT@kn@context 1255, 2834, 2866
- \MT@kn@do 1179, 1182
- \MT@kn@factor 152
- \MT@kn@factor@ 813, 1256, 1258
- \MT@kn@inh@name 1213, 1214, 1216
- \MT@kn@max 167
- \MT@kn@min 167
- \MT@kn@setname 1271,
1274, 1282, 3327, 3328, 3330, 3331
- \MT@kn@split 1184, 1188
- \MT@kn@split@val 1194, 1198
- \MT@kn@unit 152
- \MT@kn@unit@ 821, 1246
- \MT@led@kern 191
- \MT@led@unhbox@line 191
- \MT@led@mac@setup 191, 235
- \MT@let@cn . 277, 278, 563, 675, 925,
1008, 1014, 1020, 1026, 1032,
1066, 1178, 1328, 1339, 1453,
1490, 1493, 1823, 1894, 2256, 3018
- \MT@let@nc
. 276, 1469, 1496, 1499, 1772,
1787, 1811, 2258, 2274, 2495,
2663, 2675, 2752, 2759, 2901, 2904
- \MT@let@nn 278,
814, 819, 822, 834, 2296, 2856
- \MT@letterspacing 152, 1284, 3336, 3337
- \MT@letterspacing@
... 1257, 1283, 1284, 1298, 1300
- \MT@letterspacing@default .. 181, 3337
- \MT@listname 1403,
1450, 1453, 1459, 1460, 1484, 1485
- \MT@load 2150,
2172, 2197, 2218, 2257, 2258, 2363
- \MT@load@list 673, 923, 1064, 1176, 1326
- \MT@loop 481, 487, 2674
- \MT@lower 1842, 1848, 2318
- \MT@make@string . 517, 1791, 2025, 2038
- \MT@map@clist@ 392
- \MT@map@clist@c
392, 1758, 1971, 2002, 2485, 3109
- \MT@map@clist@n 392, 570, 1311, 1788,
1836, 1913, 2070, 2302, 2312,
2325, 2555, 2733, 2794, 2890, 2933
- \MT@map@tlist@ 409
- \MT@map@tlist@c
..... 409, 448, 459, 733, 991,
1110, 1215, 1407, 1700, 2630, 2689
- \MT@map@tlist@n . 409, 1386, 2977, 3028
- \MT@maybe@do . 566, 666, 917, 1054, 1166
- \MT@next@listname 1410, 1419
- \MT@next@listname@ .. 1428, 1440, 1446
- \MT@nl@setname 2058
- \MT@noligatures 554, 1305
- \MT@noligaturesfalse 139
- \MT@noligaturestrue 139, 2056
- \MT@nonselectedfalse 929, 944
- \MT@nonselectedtrue 929, 932
- \MT@norestfalse
... 1530, 1533, 1541, 1550, 1571
- \MT@noresttrue 1505, 1530, 1559
- \MT@old@cmd 29, 34, 37, 40, 43
- \MT@opt@autofalse 2884
- \MT@opt@autottrue 2884
- \MT@opt@expansionfalse 2884
- \MT@opt@expansiontrue 2884
- \MT@orig@foreign@language . 3370, 3372
- \MT@orig@pickupfont .. 257, 1676, 1714
- \MT@orig@select@language .. 3359, 3361
- \MT@pdfcprot@error 182, 234
- \MT@pdftex@no
.. 62, 100, 104, 193, 739, 956,
1053, 1165, 1264, 1305, 2054,
2460, 2466, 3063, 3225, 3250, 3298
- \MT@permute 2164,
2188, 2210, 2231, 2501, 2573
- \MT@permute@ 2573
- \MT@permute@@ 2573
- \MT@permute@@@ 2573
- \MT@permute@@@@ 2573
- \MT@permute@define .. 2610–2613, 2659
- \MT@permute@reset 2576, 2665
- \MT@permute@reset@ .. 2666–2669, 2672

- \MT@permutelist 2162, 2164, 2186, 2208, 2229, 2328, 2329, 2331, 2499, 2624, 2625, 2629, 2635, 2639, 2643–2646, 2654, 2655, 2690, 2723, 2724
- \MT@pickupfont 1711, 1716, 1719
- \MT@pr@cname 673–675, 890, 896, 2150, 2169
- \MT@pr@context 2831, 2866
- \MT@pr@do 676, 702
- \MT@pr@factor 152, 3175, 3176
- \MT@pr@factor@ 813
- \MT@pr@inh@name 731, 732, 734
- \MT@pr@level 152, 3173, 3174
- \MT@pr@max 167
- \MT@pr@min 167
- \MT@pr@setname 1978, 3178, 3179, 3181, 3182
- \MT@pr@split 704, 708
- \MT@pr@split@val 714, 718
- \MT@pr@unit 152, 3054, 3177
- \MT@pr@unit@ 821, 893
- \MT@preset@aux 898, 900, 902, 903, 906, 1251, 1253, 1260, 1261
- \MT@preset@aux@factor 898, 906, 1149–1151, 1251
- \MT@preset@aux@space 900, 906, 1153, 1253
- \MT@preset@ex 1047
- \MT@preset@kn 1241
- \MT@preset@kn@ 1241
- \MT@preset@pr 888
- \MT@preset@pr@ 888
- \MT@preset@sp 1139
- \MT@preset@sp@ 1139
- \MT@ProcessOptionsWithKV .. 3106, 3126
- \MT@protrusion 548, 666, 3185
- \MT@protrusionfalse 139, 3144
- \MT@protrusiontrue 139, 3061
- \MT@rbba@expansion 533
- \MT@rbba@kerning 533
- \MT@rbba@protrusion 533
- \MT@rbba@spacing 533
- \MT@rem@from@list 434, 1707
- \MT@rem@from@lists 1700, 1705
- \MT@remove@spaces 516, 1389
- \MT@repeat 481, 487, 2680
- \MT@reset@context 2842, 2844
- \MT@reset@ef@codes 921, 942, 956
- \MT@reset@ef@codes@ 947, 957, 961
- \MT@reset@kn@codes 1171, 1230
- \MT@reset@kn@codes@ 1230
- \MT@reset@pr@codes 668, 688
- \MT@reset@pr@codes@ 688
- \MT@reset@sp@codes 1059, 1127
- \MT@reset@sp@codes@ 1127
- \MT@restore@catcodes 127, 137
- \MT@saved@setupfont 2876, 2883
- \MT@scale 500, 742, 787, 802, 805, 978, 1257
- \MT@scale@factor 744, 763, 785, 908, 1049
- \MT@scale@to@em 721, 727, 739, 760, 914, 1090, 1097, 1104, 1156, 1160, 1202, 1209
- \MT@selectedfalse 139
- \MT@selectedtrue 139
- \MT@series 556, 1422, 1434
- \MT@set@all@ex 947, 1050
- \MT@set@all@kn 1230, 1262
- \MT@set@all@pr 679, 688, 904
- \MT@set@all@sp 1127, 1163
- \MT@set@default@set 1995, 1996, 1998, 1999, 2005, 2010
- \MT@set@ex@codes 946, 3255, 3257
- \MT@set@ex@codes@n 930, 946, 3257
- \MT@set@ex@codes@es 918, 3255
- \MT@set@ex@heirs 993, 1040
- \MT@set@ex@list 2190, 2192
- \MT@set@ex@opt 2184, 2266
- \MT@set@inh@list 2474, 2476
- \MT@set@kn@codes 1170
- \MT@set@kn@heirs 1217, 1221
- \MT@set@kn@list 2233, 2235
- \MT@set@kn@opt 2227, 2288
- \MT@set@opt@ 2262–2264, 2276–2280, 2284–2286, 2289–2291, 2294
- \MT@set@pr@codes 667
- \MT@set@pr@heirs 735, 879
- \MT@set@pr@list 2166, 2168
- \MT@set@pr@opt 2160, 2261
- \MT@set@sp@codes 1058
- \MT@set@sp@heirs 1112, 1116
- \MT@set@sp@list 2212, 2214
- \MT@set@sp@opt 2206, 2283
- \MT@setupfont 539, 1698, 2876, 2879, 2883, 3150
- \MT@setupfont@hook 220, 236, 238, 244, 252, 542
- \MT@shape 556, 1423, 1435
- \MT@shrink 152, 937, 1023, 3192, 3194, 3196, 3200, 3201, 3204, 3208, 3260, 3272
- \MT@shrink@ 927, 937, 943, 1006

- \MT@shrink@default 177, 3196
 \MT@size .. 466, 470, 471, 556, 655, 658
 \MT@size@name 457, 1450
 \MT@sp@cc@name 1064–
 1066, 1141, 1147, 2197, 2215
 \MT@sp@context 2833, 2866, 3303
 \MT@sp@do 1067, 1070
 \MT@sp@factor 152
 \MT@sp@factor@ 813
 \MT@sp@inh@name 1108, 1109, 1111
 \MT@sp@max 167
 \MT@sp@min 167
 \MT@sp@setname 3313, 3314, 3316, 3317
 \MT@sp@split 1072, 1076
 \MT@sp@split@val 1082, 1086
 \MT@sp@unit 152
 \MT@sp@unit@ 821, 1144
 \MT@spacing 551, 1053, 3320, 3347
 \MT@spacingfalse 139, 3147, 3346
 \MT@spacingtrue 139
 \MT@split@name 543, 556
 \MT@split@val 1086
 \MT@step 152,
 938, 1029, 3199, 3215, 3222, 3273
 \MT@step@ 927, 938, 943, 1006
 \MT@step@default 177
 \MT@stretch 152, 936,
 1017, 3189, 3190, 3193, 3194,
 3200, 3202, 3207, 3210, 3259, 3272
 \MT@stretch@ 927, 936, 943, 1006
 \MT@stretch@default 177, 3190
 \MT@temp . 1926, 1933, 1948, 2339, 2347
 \MT@tempencoding 2614, 2621
 \MT@tempfamily 2615
 \MT@tempseries 2616
 \MT@tempshape 2617
 \MT@tempsize
 ... 2618, 2628, 2631, 2636, 2670
 \MT@test@ast 1799, 1816
 \MT@testrest 1531
 \MT@tlist@break
 ... 409, 454, 478, 1412, 2726
 \mt@toks
 139, 774, 780, 808, 1002, 1506,
 1539, 1546, 1592, 1651, 1663, 1669
 \MT@try@order 1407, 1415
 \MT@upper 1842, 1848, 2318
 \MT@use@set 1964,
 1965, 1967, 1968, 1974, 1978
 \MT@val 1789, 1791, 1793,
 1799, 1801, 1806, 1837–1839,
 1851, 1854, 1856, 1859, 1861,
 1862, 1870, 1872, 1873, 1875,
 1878, 1880, 1891, 1894, 1898,
 1902–1904, 1906, 1908, 1914,
 1915, 1918, 1934, 1936, 1939,
 1940, 1944, 1946, 1951, 2303,
 2305, 2313–2315, 2326, 2327,
 2332, 2348, 2350, 2516, 2517,
 2519, 2524, 2554, 2561, 2567,
 2568, 2734, 2736, 2738, 2745,
 2749, 2756, 2764, 2795, 2797,
 2799, 2805, 2810, 2852–2855,
 2891, 2892, 2895, 2897, 2900,
 2903, 2906, 2907, 2912, 2934,
 2935, 2938, 2940, 2943, 2944, 2949
 \MT@vinfo 45, 56,
 540, 583, 816, 825, 829, 863,
 1009, 1015, 1021, 1027, 1034,
 1322, 1334, 1356, 1363, 1366,
 1462, 1464, 1471, 2041, 2046,
 2845, 2853, 2997, 3006, 3363, 3374
 \Mt@warn@ascii 1548, 1650
 \MT@warn@code@too@large . 792, 796, 800
 \MT@warn@err 45, 3003, 3004
 \MT@warn@ex@too@large ... 981, 984, 999
 \MT@warn@missing@char ... 751, 756, 771
 \MT@warn@number@too@large . 1575, 1656
 \MT@warn@unknown 1522, 1667
 \MT@warn@unknown@i 1525, 1661
 \MT@warning 30, 45, 1290,
 1330, 1342, 1473, 1770, 1874,
 1906, 1932, 1947, 1987, 2016,
 2027, 2048, 2062, 2243, 2346,
 2398, 2407, 2419, 2430, 2447,
 2463, 2525, 2622, 2644, 2723,
 2763, 2782, 2809, 3003, 3047, 3096
 \MT@warning@nl .. 45, 103, 207, 212,
 246, 692, 773, 780, 806, 856,
 894, 1000, 1145, 1247, 1651,
 1657, 1662, 1668, 1678, 1826,
 2269, 2911, 2948, 2968, 2989,
 3004, 3008, 3022, 3037, 3056,
 3084, 3102, 3141, 3217, 3248,
 3240, 3261, 3304, 3341, 3349, 3390
 \MT@while@num 487,
 682, 950, 1130, 1233, 2247, 2505
 \MT@xadd 378, 1273, 1357,
 1364, 1367, 1691, 1699, 1792,
 1840, 1916, 2043, 2316, 2560, 2838
 \MT@xaddb 385, 2330, 2634
- N
- ngerman (package) 22

- `\normalfont` 265, 3282, 3290
`\nullfont` 265
- P**
- `\pdfadjustinterwordglue` ... 2821, 3300
`\pdfadjustspacing` ... 2789, 3269, 3287
`\pdfappendkern` 1280, 2823, 3325
`pdfcpot` (package)
 4, 18, 19, 28, 41, 63, 147
`\pdffontexpand` 927, 943
`\pdfnoligatures` 1321
`\pdfoutput`
 66, 3062, 3128, 3131, 3138, 3226
`\pdfprependkern` 1281, 2822, 3324
`\pdfprotrudechars` 2788, 3173
`\pdftexrevision` 65, 70, 84, 86
`\pdftexversion` 64, 68–70, 79, 81, 83, 91
`\pickup@font` 257, 1677, 1679, 1684,
 1711, 1714, 1716, 1719, 1722, 1727
`pifont` (package) 64
`protrusion` (option) 5, 10, 91, 92
`PSNFSS` (package) 141
`pstricks` (package) 7, 96
`pxfonts` (package) 16, 74
- Q**
- `qfonts` (package) 16, 150
- R**
- `relsize` (package) 69
`\rightmarginkern` 200
`\rPCODE` 684, 728, 729, 881, 885
- S**
- `selected` (option)
 7, 13, 14, 22, 46, 58, 92, 148
- `\selectfont` 2841, 2846
`\set@fontsize` 1898
`\SetExpansion` 7, 13, 2172, 3159
`\SetExtraKerning` 15, 2218, 3162
`\SetExtraSpacing` 14, 2197, 3161
`\SetProtrusion` 6, 11, 2150, 3158
`\sfcode` 3302
`\shbrcode` . 1105, 1106, 1119, 1124, 1133
`\showhyphens` 3280, 3284
`shrink` (option) 6, 7, 14, 22, 93, 98, 150
`slantsc` (package) 122
`soul` (package) 143
`spacing` (option) 92
`statex` (package) 29
`\stbrcode` . 1098, 1099, 1118, 1123, 1132
`step` (option) ... 7, 22, 93, 98, 147, 150
`stretch` (option) . 6, 7, 14, 18, 22, 93, 98
- T**
- `\textls` 17, 260, 1296
`tifa` (package) 64
`txfonts` (package) 16, 74
- U**
- `unit` (option) 6, 13, 21, 149, 150
`\UseMicrotypeSet`
 .. 3, 10, 27, 38, 1746, 1961, 3157
- V**
- `verbose` (option) 8, 21, 93, 148, 149
- Z**
- `zefonts` (package) 16, 74