

The microtype package

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

R Schlicht
w.m.l@gmx.net

v2.3b
2008/06/04

Abstract

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

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

The alternative package `letterspace`, which also works with plain \TeX , provides the user commands for letterspacing only, omitting support for all other extensions (see section 7).

This package is copyright © 2004–2008 R Schlicht. It may be distributed and/or modified under the conditions of the [\$\LaTeX\$ Project Public License](#), either version 1.3c of this license or (at your option) any later version. This work has the LPPL maintenance status ‘author-maintained’.

¹ Currently, this package provides protrusion settings for Computer Modern Roman, Palatino, Times, URW Garamond, Adobe Garamond and Minion, Bitstream Charter and Letter Gothic, the AMS symbols and Euler fonts, for various Euro symbol fonts, as well as some generic settings for unknown fonts (cf. table 3 on page 21). Contributions are very welcome.

Contents

1	Micro-typography with pdf\TeX	4
2	Getting started	5
3	Options	6
3.1	Enabling the micro-typographic features	6
3.2	Character protrusion	7
3.3	Font expansion	8
3.4	Tracking/letterspacing	9
3.5	Miscellaneous options	9
3.6	Changing options later	10
4	Selecting fonts for micro-typography	10
5	Micro fine tuning	12
5.1	Character protrusion	13
5.2	Font expansion	14
5.3	Tracking	16
5.4	Interword spacing	18
5.5	Additional kerning	19
5.6	Character inheritance	20
5.7	Configuration files	20
6	Context-sensitive setup	22
7	Letterspacing revisited	24
8	Disabling ligatures	25
9	Hints and caveats	25
10	Contributions	28
11	Acknowledgments	28
12	References	29
13	Short history	29
14	Implementation	33
14.1	Preliminaries	34
	[Debugging – 35] [Requirements – 37] [Declarations – 39] [Auxiliary macros – 40] [Compatibility – 48]	
14.2	Font Setup	52
	[Protrusion – 57] [Expansion – 64] [Interword spacing (glue) – 66] [Additional kerning – 68] [Tracking – 69] [Disabling ligatures – 77] [Loading the configuration – 78] [Translating characters into slots – 82] [Hook into \LaTeX 's font selection – 87] [Context-sensitive setup – 90]	

14.3	Configuration	92
	[Font sets – 92] [Variants and aliases – 98] [Disabling ligatures – 99]	
	[Interaction with babel – 99] [Fine tuning – 99] [Character inheritance – 105] [Permutation – 107]	
14.4	Package options	111
	[Declaring the options – 111] [Reading the configuration file – 115] [Hook for other packages – 116] [Changing options later – 116] [Processing the options – 118]	
15	Configuration files	126
15.1	Font sets	126
15.2	Font variants and aliases	127
15.3	Interaction with babel	128
15.4	Note on admissible characters	128
15.5	Character inheritance	129
	[OT1 – 129] [T1 – 129] [LY1 – 130] [OT4 – 131] [QX – 131] [T5 – 132]	
	[Euro symbols – 132]	
15.6	Tracking	133
15.7	Font expansion	133
15.8	Character protrusion	134
	[Default – 135] [Italics – 143] [Small caps – 151] [Italic small caps – 154]	
	[Text companion – 155] [Computer Modern math – 160] [AMS symbols – 164] [Euler – 168] [Euro symbols – 172]	
15.9	Interword spacing	173
	[Nonfrenchspacing – 174]	
15.10	Additional kerning	175
	[French – 176] [Turkish – 176]	
16	Auxiliary file for micro fine tuning	177
A	Change history	179
B	Index	185
C	The L^AT_EX Project Public License	193

List of Tables

1	Availability of micro-typographic features	7
2	Predefined font sets	12
3	Fonts with tailored protrusion settings	21
4	Order for matching font attributes	80

1 Micro-typography with pdf_T_EX

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

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

Protrusion	off
Expansion	off

Both features are enabled throughout this document.

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

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

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

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

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

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

Robust and hyphenatable *letterspacing* (tracking) has always been extremely difficult to achieve in _T_EX. Although the `soul` package undertook great efforts in making this possible, it could still fail in certain circumstances; even to adjust the tracking of a font throughout the document remained impossible. Employing pdf_T_EX's new extension, this no longer poses a problem. The `microtype` package

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

Adjustment of interword spacing is based upon the idea that in order to achieve a uniform greyness of the text, the space between words should also depend on the surrounding characters. For example, if a word ends with an ‘r’, the following space should be a tiny bit smaller than that following, say, an ‘m’. You can think of this concept as an extension to T_EX’s ‘space factors’. However, while space factors will influence all three parameters of interword space (or glue) by the same amount – the kerning, the maximum amount that the space may be stretched and the maximum amount that it may be shrunk – pdfT_EX provides the possibility to modify these parameters independently from one another. Furthermore, the values may be set differently for each font. And, probably most importantly, the parameters may not only be increased but also decreased. This feature may enhance the appearance of paragraphs even more. Emphasis in the last sentence is on the word ‘may’: this extension is still highly experimental – in particular, only ending characters will currently have an influence on the interword space. Also, the settings that are shipped with microtype are but a first approximation, and I would welcome corrections and improvements very much. I suggest reading the reasoning behind the settings in section 15.9.

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

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

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

2 Getting started

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. If this, on the other hand, does not satisfy you – be it for theoretical or practical reasons – this manual will guide you on the path to the desired results along the following milestones:

- Enable the respective micro-typographic feature, either via the respective package option or with the `\microtypesetup` command (section 3).
- Select the fonts to which this feature should be applied by declaring and activating ‘sets of fonts’. Some sets are predefined, which may be activated directly in the package options (section 4).
- Fine-tune the micro-typographic settings of the fonts or sets of fonts (section 5).
- If you’re of the kind who always wants to march on, you’ll certainly be interested in the possibility of context-sensitive setup (section 6).
- You are even countenanced to leave the path of typographic virtue and steal some sheep (section 7) or trespass in other ways (section 8).
- Should you encounter any obstacles, follow the hints and caveats (section 9).

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 and/or the output mode).

3.1 Enabling the micro-typographic features

protrusion true, false, compatibility, nocompatibility, ** *true

expansion These are the main options to control the level of micro-typographic refinement which the fonts in your document should gain. By default, the package is moderately greedy: character protrusion will be enabled, font expansion will only be disabled in circumstances where 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 3.5). In other words, `microtype` will try to apply as much micro-typography as can safely be expected to work under the respective conditions (and it is usually not necessary to load the package with different options for PDF resp. DVI mode).

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

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

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

```
\usepackage{microtype}
```

Table 1: Availability of micro-typographic features

pdfTeX		Micro-typographic features					
version	output	Protrusion	Expansion	(= auto)	Kerning	Spacing	Tracking
< 0.14f	DVI/PDF	∅	∅	∅	∅	∅	∅
≥ 0.14f	DVI/PDF	★	☒	∅	∅	∅	∅
≥ 1.20	DVI	★	☒	∅	∅	∅	∅
	PDF	★	★	★	∅	∅	∅
≥ 1.40	DVI	★	☒	∅	☒	☒	∅
	PDF	★	★	★	☒	☒	☒ ^a
★ = enabled ☒ = not enabled ∅ = not available							^a 1.40.4 recommended

When pdfTeX employs font expansion and character protrusion, line breaks (and consequently, page breaks) may turn out differently. If this is not desired – because you are re-typesetting a book whose pagination must not change – you may pass the value `compatibility` to the protrusion and/or expansion options. Typographically, however, the results will be suboptimal, hence the default value is `nocompatibility`.

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

tracking

spacing

kerning

true, false, **

There is no compatibility level for the new extensions of tracking, interword spacing, and additional kerning. Therefore, they can only be switched on or off, or they may be activated by passing a set name to the option. By default, neither feature is enabled.

false

In table 1, you find an overview of which micro-typographic features are available and enabled by default for the relevant pdfTeX versions and output modes.

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

3.2 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, $\langle dimension \rangle$ character
 This option is described in section 5.1, apropos the command `\SetProtrusion`. Use with care.

3.3 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 $\langle font\ name \rangle \pm \langle expansion\ value \rangle$, e. g., `cmr12+10`, as described in the [pdfTeX manual](#)).

Automatic font expansion does not work with bitmap fonts. Therefore, if you are using the Computer Modern Roman fonts in T1 encoding², you should either install the cm-super fonts or use the Latin Modern fonts (package `lmodern`).

stretch $\langle integer \rangle$ 20

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

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

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

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

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

```
\usepackage{microtype}
```

selected true, false false

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

² En passant, it may be noted that Type 1 format and T1 encoding are in no other way related than that both start with a 'T' and end with a '1'.

3.4 Tracking/letterspacing

letterspace *<integer>* 100

This option changes the default amount for tracking (see section 5.3) resp. letter-spacing (see section 7). The amount is specified in thousandths of 1em; admissible values are in the range of -1000 to $+1000$.

3.5 Miscellaneous options

DVIoutput true, false * false

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

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

When generating DVI files, font expansion has to be enabled explicitly. Neither `letterspacing` nor *automatic* font expansion will work because the postprocessing drivers (`dvips`, `dvipdfm`, etc.) resp. the DVI viewer are not able to generate the fonts on the fly.

draft true, false false

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

verbose true, false, errors, silent false

Information on the settings used for each font will be written into the log file if you enable the `verbose` option. 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. If on the other hand you have investigated all warnings and decide to ignore them, you may silence `microtype` with `verbose=silent`.

babel true, false false

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

config *<file name>* microtype

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

³ Recent \TeX systems are using pdf \TeX as the default engine even for DVI output.

3.6 Changing options later

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

Inside the preamble, this command accepts all package options described above (except for `config`). In the document body, this command may be used to change the general settings of the micro-typographic extensions. It then accepts all options from section 3.1: `expansion`, `protrusion` and `activate`, which in turn may receive the values `true`, `false`, `compatibility` or `nocompatibility`, and `tracking`, `spacing` and `kerning` with the admissible values `true` or `false`. Passing the name of a font set is not allowed. Using this command, you could for instance temporarily disable font expansion by saying:

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

4 Selecting fonts for micro-typography

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

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

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

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

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

If you now call

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

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

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

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

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

If a value is followed by an asterisk (like ‘rm*’ and ‘sf*’ in the first example), it does not designate an NFSS code, but will expand to the document’s $\langle value \rangle$ default, e.g., $\backslash rmdefault$. A single asterisk means $\langle attribute \rangle$ default, e.g., $\backslash encodingdefault$, respectively $\backslash normalsize$ for the size axis. Sizes may either be specified as a dimension (‘10’ or ‘10pt’), or as a size selection command *without* the backslash. You may also specify ranges (e.g., ‘small-Large’); while the lower boundary is included in the range, the upper boundary is not. Thus, ‘12-16’ would match 12pt, 13.5pt and 15.999pt, e.g., but not 16pt. You are allowed to omit the lower or upper bound (‘-10’, ‘large-’).

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

```
\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. A single asterisk is equivalent to ‘*/*/*/*/’, i.e., the normal font. Size selection commands are possible, too, however, ranges are not allowed.

Table 2 lists the nine predefined font sets. They may also be activated by passing their name to the feature options protrusion, expansion, tracking, spacing and kerning when loading the package, for example:

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

$\backslash UseMicrotypeSet$ [$\langle features \rangle$] { $\langle set name \rangle$ }

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

$\backslash DeclareMicrotypeSetDefault$ [$\langle features \rangle$] { $\langle set name \rangle$ }

If a feature is enabled but no font set has been chosen explicitly, the sets declared

Table 2: Predefined font sets

Set name	Font attributes				
	Encoding	Family	Series	Shape	Size
all	Ø	Ø	Ø	Ø	Ø
alltext (allmath)	Text encodings, TS1 (OML, OMS, U)	Ø	Ø	Ø	Ø
basictext (basicmath)	Text encodings (OML, OMS)	\rm*, \sf*	\md*	Ø	\normalsize, \footnotesize, \small, \large
smallcaps	Text encodings	Ø	Ø	sc	Ø
footnotesize	Text encodings, TS1	Ø	Ø	Ø	-\small
scriptsize	Text encodings, TS1	Ø	Ø	Ø	-\footnotesize
normalfont	\encoding*	\family*	\series*	\shape*	\normalsize
‘Text encodings’ = OT1, T1, LY1, OT4, QX, T5					‘...*’ = ‘...default’

by this command will be activated. By default, the ‘alltext’ font set will be used for character protrusion and additional kerning, the ‘basictext’ set for font expansion and interword spacing, and the ‘smallcaps’ set for tracking.

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

5 Micro fine tuning

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

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

The set of fonts to which the settings should apply is declared using the same syntax of *{font axis} = {value list}* pairs as for the command `\DeclareMicrotypeSet` (see section 4). The only difference is that asterisked values will be expanded immediately instead of at the end of the preamble.

To find the matching settings for a given font the package will try all combinations of font encoding, family, series, shape and size, with decreasing significance in this order. For instance, if both settings for the current family (say, T1/cmr//) and settings for italic fonts in the normal weight (T1//m/it/) exist, those for the Computer Modern Roman font would apply.⁴ The encoding must always match.

⁴ For the interested, table 4 on page 80 presents the exact order.

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 T1 encoded Computer Modern Roman family.

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

The characters may be specified either as a single character (‘A’), as a text symbol command (‘\textquoteleft’), or as a slot number: three digits for decimal notation, prefixed with ‘#’ for hexadecimal, with ‘o’ for octal (e. g., the ‘fl’ ligature in T1 encoding: 029, #1D, o35). 8-bit (and even UTF-8) characters may be entered directly or in L^AT_EX’s traditional 7-bit notation: both ‘\’A and ‘A are valid, provided the character is actually declared in both the input and the font encoding. Note that you also have the possibility to declare lists of characters that should inherit settings (see section 5.6).

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

Options:

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

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

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

factor This option can be used to influence all protrusion factors of the list, overriding any global factor setting (see section 3.2). For instance, if you want

fonts in larger sizes to be protruded less, you could load the normal lists, just with a different factor applied to them:

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

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

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

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

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

5.2 Font expansion

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

By default, all characters of a font are allowed to be stretched or shrunk by the same amount. However, it is also possible to limit the expansion of certain characters if they are more sensitive to deformation. This is the purpose of the `\SetExpansion` command. Note that it will only have an effect if the package was loaded with the selected option (cf. section 3.3). Otherwise, the expansion settings will be ignored – unlike the options in the optional first argument, which will still be evaluated.

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

The *expansion settings* consist of *<character> = <expansion factor>* 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.

⁵ The `unit` option can even be passed globally to the package (cf. section 3.2). However, all provided settings are created under the assumption that the values are relative to the character width. Therefore, you should only change it if you are certain that the default settings will not be used in your document.

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

Options:

name, load, preset, inputenc, context Analogous to `\SetProtrusion`, the optional argument may be used to assign a name to the list, to load another list, to preset all expansion factors, to set the input encoding, or to determine the context of the list (expansion contexts are only possible with pdfTeX version 1.40.4 or newer).

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

As a practical example, suppose you have a paragraph containing a widow that could easily be avoided by shrinking the font a little bit more. In conjunction with the context option (see section 6 for further details), you could thus allow for more expansion in this particular paragraph:

```
\SetExpansion
[ context = sloppy,
  stretch = 30,
  shrink   = 60,
  step     = 5 ]
{ encoding = {OT1,T1,TS1} }
{ }
% ... END PREAMBLE
{\microtypecontext{expansion=sloppy}%
This paragraph contains an 'unnecessary' widow.}
```

This method of employing contexts to temporarily apply different expansion parameters only works with pdfTeX version 1.40.4 or later (for older versions, a dirty trick is laid out in section 14.2 on page 54). Also note that pdfTeX prohibits the use of fonts with different expansion limits or steps (even of different fonts) within one paragraph, hence the sloppy context has to be applied to complete paragraphs.

factor This option provides a different method to alter expansion settings for certain fonts, working around the restriction just mentioned. 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.

5.3 Tracking

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

An important typographic technique – which was missing in T_EX for a long time – is the adjustment of tracking, i. e., the uniform addition or subtraction of letter space to/from all the characters in a font. For example, it is good typographic practice to slightly space out text set in all capitals or small capitals (as in this document). Legibility may also be improved by minimally increasing the tracking of smaller and decreasing that of larger type.⁶ The `\SetTracking` command allows to specify the tracking amount for different fonts or font sets. It will also be evaluated by the `\textls` command, which may be used for letterspacing shorter pieces of text (see section 7).

The *tracking amount* is specified in thousandths of 1em (or the given unit); negative values are allowed, too.

Options:

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

spacing When the inter-*letter* spacing is altered, the inter-*word* spacing probably also needs to be adjusted. This option expects three numbers for interword space, stretch and shrink respectively, which are given in thousandths of 1em (or of the current unit). If a value is followed by an asterisk, it denotes thousandths of the respective font dimension which will be added to it. For instance, with

```
\SetTracking[ spacing = {25*,166, } ]{ encoding = *, shape = sc }{ 25 }
```

the interword space will be increased by 2.5%, the stretch amount will be set to 0.166em, while the shrink amount will be left untouched. If you don't specify the spacing option, the interword space will be scaled by the current letterspace amount (as in the above example), while stretch and shrink will not be changed.

outer spacing If an interword space immediately precedes or follows letter-spaced text, it will by default be equal to that within the text. With this option, which accepts the same values as spacing, it may be adjusted independently.

outer kerning If, on the other hand, no interword space precedes or follows, you may still want to slightly set off the first and last letter from adjoining letters. This option expects the kerning amounts for left and right hand side, separated by a comma, in thousandths of 1em (or the current unit). If a value is followed by an asterisk, it denotes thousandths of the current letterspacing amount. A single asterisk means '500*'; this is also the default, i. e., the sum of the outer kerns is by

⁶ With full-featured fonts like Computer Modern, this is usually not necessary, though, since they come in optical sizes, and the tracking of the small-capitals font is already adjusted.

default equal to the current letterspace amount. To remove kerning on both sides, you would write ‘outer kerning={0,0}’.

no ligatures As far as pdf \TeX is concerned, ligatures in letterspaced fonts would be constructed as usual, which may be advisable when changing the tracking by only a small amount. For larger letterspacing amounts, on the other hand, the normal letter space within ligatures would have displeasing effects. This key expects a comma-separated list of characters for which ligatures should be disabled; only the character that begins a ligature must be specified. If the key is given without a value, *all* ligatures of the font will be disabled. This is not recommended, however, since it also entails that kerning will be switched off.⁷ The default settings disable ligatures for the character ‘f’ only, i.e., ‘ff’, ‘fi’, ‘ffi’, etc.⁸ In exceptional situations, you can manually break up a ligature by inserting ‘{\kern0pt}’ resp. babel’s “| shortcut, or protect it by enclosing it in \lslig (see section 7).

Since a picture is worth a thousand words, probably even more if, in our case, it depicts a couple of letterspaced words, let’s bring one to sum up these somewhat confusing options. Suppose you had the following settings (which I would in no way recommend; they are only for illustrative purposes):

```
\SetTracking
[ no ligatures = {f},
  spacing      = {600*,-100*, },
  outer spacing = {450,250,150},
  outer kerning = {*,*} ]
{ encoding = * }
{ 160 }
```

and then write:

```
Stop \textls{stealing sheep}!
```

this is the (typographically dubious) outcome:



While the word ‘Stop’ is not letterspaced, the space between the letters in the other two words is expanded by the *tracking amount* of $160/1000\text{em} = 0.16\text{em}$. The *inner space* within the letterspaced text is increased by 60%, while its *stretch* amount is decreased by 10% and the *shrink* amount is left untouched. The *outer space* (of 0.45em) immediately before the piece of text may *stretch* by 0.25em and *shrink* by 0.15em. Note that there is no outer space after the text, since the exclamation mark immediately follows; instead, the default *outer kern* of half the letterspace amount (0.08em) is added. Furthermore, one *ligature* wasn’t broken up, because we neglected to specify the ‘s’ in the `no ligatures` key.

⁷ The inseparable connexion of ligatures and kerns is a limitation of \TeX that will not be lifted before the advent of Lua \TeX .

⁸ With pdf \TeX versions older than 1.40.4, *all* ligatures, and hence all kerning, will be disabled. It is therefore recommended to use at least version 1.40.4.

Click on the image to show the kerns and spacings involved. Click on emphasised words in the text below to reveal the relation of image and code.

As another, more realistic example, suppose you want to space out all small capitals by 50/1000em, fonts smaller than `\small` by 0.02em, and to decrease the tracking of large type by 0.02em. You can achieve this with the following settings:

```
\usepackage[tracking=true]{microtype}
\DeclareMicrotypeSet*[tracking]{my}
{ encoding = *,
  size     = {-small,Large-},
  font     = */*/*/sc/* }
\SetTracking[ no ligatures = f ]{ encoding = *, shape = sc}{ 50 }
\SetTracking{ encoding = *, size = -small }{ 20 }
\SetTracking{ encoding = *, size = Large- }{ -20 }
```

Letterspaced fonts for which settings don't exist will be spaced out by the default of 0.1em (adjustable with the package option `letterspace`, see section 3.5). Suppose your editor wants you to shorten your 1000 pages chef-d'œuvre by a handful of pages, you could load `microtype` with (fingers crossed):

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

5.4 Interword spacing

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

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

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

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

Options:

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

unit You can specify the unit by which the specified numbers are measured. Possible values are: `character`, a *⟨dimension⟩* and, additionally, `space`. The latter will measure the values in thousandths of the respective space dimension set by the font. By default, the unit is measured by the space dimensions. For example, with these (nonsensical) settings:

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

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

5.5 Additional kerning

`\SetExtraKerning` [*⟨options⟩*] {*⟨set of fonts⟩*} {*⟨kerning settings⟩*}

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

I should not neglect to mention a limitation of this additional kerning: words *immediately following* such a kern (not separated by a space) will not be hyphenated, unless you insert the breakpoints manually, e. g., for kerning after the apostrophe, ‘l 'apostrophe’. This restriction of pdfTeX will hopefully be lifted soon.

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

Options:

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

unit Admissible values are: `space`, `character` and a *⟨dimension⟩*. By default, the values denote thousandths of 1em.

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

For example, you can find the following settings, intended to be used for documents written in French, in the main configuration file:

```
\SetExtraKerning
[ name      = french-default,
  context   = french,
  unit      = space ]
{ encoding = {OT1,T1,LY1} }
{
  : = {1000,}, % = \fontdimen2
  ; = {500, }, % ~ \thinspace
  ! = {500, },
  ? = {500, }
}
```

What is the result of these settings? If they are active, like in the current paragraph, a thin space will be inserted in front of each question mark, exclamation mark and semicolon; a normal space in front of the colon. Read section 6 to learn how to activate these settings! This paragraph was input like this :

```
\begin{microtypecontext}{\kerning=french}
What is the result of these settings? If they are active, like in the
current paragraph, a thin space will be inserted in front of each
question mark, exclamation mark and semicolon; a normal space in front
of the colon. Read section~\ref{sec:context} to learn how to activate
these settings! This paragraph was input like this:
\end{microtypecontext}
```

5.6 Character inheritance

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

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

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

5.7 Configuration files

The default configuration, consisting of inheritance settings, declarations of font sets and alias fonts, and generic protrusion, expansion, spacing and kerning settings, will

Table 3: Fonts with tailored protrusion settings

Font family (NFSS code)	Features	
	Encodings	Shapes
Generic	OT1, T1, LY1, QX, (TS1) ^a	n, (it, sl, sc) ^a
Computer Modern Roman (cmr) ^b	OT1, OT4, T1, T5, LY1, TS1	n, it, sl, sc
Bitstream Charter (bch) ^c	OT1, T1, T5, LY1, TS1	n, it, (sl) ^d , sc
Adobe Garamond (pad, padx, padj)	OT1, T1, LY1, TS1	n, it, (sl) ^d , sc
URW Garamond (ugm) ^e	OT1, T1, TS1	n, it
Bitstream Letter Gothic (blg) ^f	OT1, T1, TS1	n, it
Adobe Minion (pmnx, pmnj) ^g	OT1, T1, LY1, TS1	n, it, (sl) ^d , sc, si
Palatino (ppl, pplx, pplj) ^h	OT1, OT4, T1, LY1, (TS1) ^a	n, it, (sl) ^d , sc
Times (ptm, ptmx, ptmj) ⁱ	OT1, OT4, T1, LY1, QX, (TS1) ^a	n, it, (sl) ^d , sc
Computer Modern math (cmsy, cmm)	OML/OMS	n/it
AMS symbols (msa, msb)	U	n
Euler (eur, eus, euf) ^j	U	n
Euro symbols (Adobe, ITC, marvosym)	U/OT1	n, it

^a Incomplete
^b Aliases: Latin Modern (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor)
^c Aliases: mathdesign/Charter (mdbch), MicroPress's chmath (chr)
^d Settings inherited from italic shape
^e Alias: mathdesign/URW Garamond (mdugm)
^f Alias: ulgothic (ulg)
^g By courtesy of Harald Harders (h.harders@tu-bs.de)
^h Aliases: pxfonts (pxr), qfonts/QuasiPalatino, T_EX Gyre Pagella (qpl), FPL Neu (fp9x, fp9j)
ⁱ Aliases: txfonts (txr), qfonts/QuasiTimes, T_EX Gyre Termes (qtm)
^j Alias: eulervm (zeur, zeus)

be loaded from the file `microtype.cfg`. You may extend this file with custom settings (or load a different configuration file with the ‘`config`’ option, see section 3.5).

If you are embarking on creating new settings for a font family, you should put them into a separate file, whose name must be: ‘`mt-⟨font family⟩.cfg`’ (e. g., ‘`mt-cmr.cfg`’), and may contain all commands described in the current section 5. These files will be loaded automatically if you are actually using the respective fonts. This package ships with configuration files for a number of font families. Table 3 lists them all.

`\DeclareMicrotypeVariants` {⟨list of suffixes⟩}

`\DeclareMicrotypeVariants*` On its search for a configuration file, the package will also try to remove from the font name a suffix of one or more letters that denotes a ‘variant’ of the base font (cf. Karl Berry’s [Fontname](#)). This allows it to put settings for, e. g., the fonts `padx` (expert set), `padj` (oldstyle numerals) and `pad` (plain) into one and the same file `mt-pad.cfg`. This command expects a comma-separated list of variant suffixes. The starred version appends the suffix(es) to the existing list. The default declaration in `microtype.cfg` is:

```
\DeclareMicrotypeVariants{x,j,w,a,d,0,1}
```

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

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

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

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

`\LoadMicrotypeFile` {**}

In rare cases, it might be necessary to load a font configuration file manually, for instance, from within another configuration file, or to be able to extend settings defined in a file that would otherwise not be loaded automatically, or would be loaded too late.⁹ This command will load the file `mt-.cfg`.

6 Context-sensitive setup

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

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

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

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

`\end{microtypecontext}` Like many \LaTeX commands, it is also available in the form of an environment.

`\textmicrotypecontext` {*<context assignments>*} {*<general text>*}

As another possibility, the command `\textmicrotypecontext` sets the context(s) for the text given in the second argument.

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

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

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

and have the context changed in the footnote marker command. This command differs among the various classes; for the base classes, e.g., `article`, it would be:

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

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

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

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

```
\microtypecontext{kerling=french}
```

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

```
\textmicrotypecontext{kerling=}{English text!}
```

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

Instead of adding these commands manually to your document, you may also load `microtype` with the `babel` option (see section 3.5). The current language will then be automatically detected and the contexts set accordingly.

`\DeclareMicrotypeBabelHook` $\{\langle list\ of\ babel\ languages \rangle\} \{\langle context\ list \rangle\}$

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

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

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

7 Letterspacing revisited

`\textls` [*amount*] {*general text*}

`\textls*` While the tracking feature, described in section 5.3, will apply to sets of fonts, you may also want to letterspace shorter pieces of text, regardless of the font in which they are typeset.¹⁰ For such ad-hoc letterspacing, microtype introduces two commands that can be used in the same way as L^AT_EX's text commands (independently of whether the tracking option is enabled): `\textls` expects the text in the mandatory argument, while `\lsstyle` will switch on letterspacing for all subsequent fonts until the end of the current group. The starred version of `\textls` does not add any extra kerning before or after the text, which may be useful, e. g., for section titles. By default, each character will be spaced out by $100/1000\text{em} = 0.1\text{em}$; this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option, with decreasing significance in this order.

`\lslig` {*ligature*}

Since the commands `\textls` and `\lsstyle` will also evaluate the ‘no ligatures’ key for the respective font, you need not worry about protecting or breaking ligatures with most fonts. However, in certain situations, there may be a conflict of ligatures beginning with the same letter, where some of them should be inhibited, while others should not. When letterspacing text typeset in Fraktur fonts, for example, the ligatures ‘ch’, ‘ck’, ‘tz’ and ‘sz’ (‘ß’) should never be broken up; you also usually see the ‘st’ (‘ſt’) ligature in letterspaced text. Furthermore, at least the `yfonts` package realises the short s (‘ſ’) as the ligature ‘s:’. On the other hand, the ‘ct’ ligature and the other ‘long s’ ligatures often found in Fraktur fonts should be suppressed. There are two ways to solve this problem: either don’t disable the ‘s’ and/or ‘c’ ligatures and break those that need to be broken up by inserting ‘`\kern0pt`’ or babel’s “|” shortcut; or disable them and protect those ligatures that need to be protected by enclosing them in the `\lslig` command. So, the following two solutions have the same result (namely, ‘A^usſi^{ch}tſloſigkeit’).

```
\SetTracking[no ligatures={f}]{encoding = LY, family = yfrak}{}
\textfrak{\lsstyle Aus:s{\kern0pt}ichſt:loſ{\kern0pt}igkeit}
```

```
\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{}
\textfrak{\lsstyle Au\lslig{s:}si\lslig{ch}t\lslig{s:}loſigkeit}
```

`letterspace.sty` These three commands (plus the `letterspace` option, described in section 3.4) are also available with the alternative `letterspace` package, which is in fact a much stripped-down version of `microtype`, omitting support for all the other extensions (and also omitting the possibilities of the `\SetTracking` command – all ‘f’ ligatures will be disabled, inner and outer spacing and outer kerning will be set to the default

¹⁰ Letterspacing should be used cautiously; in particular, letterspacing lower-case text is held in abhorrence by honourable typographers. Unless you know what you are doing, you should probably only letterspace small-capitals or all-capitals. Another just cause may be emphasis in texts typeset in Fraktur fonts.

values described in section 5.3). If you prefer to forgo microtype’s specialties, you may load the `letterspace` package instead. Both packages should not be used at the same time.

In contrast to `microtype`, which requires \LaTeX , the `letterspace` package also works with `eplain` or even only `miniltx`: for use with `eplain`, load the package with `\usepackage` inside the `\beginpackages ... \endpackages` environment; with `miniltx` (which does not support package options) simply `\input letterspace.sty`.

8 Disabling ligatures

`\DisableLigatures` [*<characters>*] {*<set of fonts>*}

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

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

It is also possible to disable selected ligatures only. The optional argument may contain a comma-separated list of characters for which the ligature mechanism should be inhibited:

```
\DisableLigatures[?,!]{encoding = T1} % inhibit ?' !', but not fi – » etc.
```

The character that begins the ligature(s) is what matters. This command may only be used in the preamble, and only once. It requires pdf \TeX 1.30 or newer.

9 Hints and caveats

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

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

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

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

Settings for Cyrillic/Greek/Thai etc. encodings are not yet included. The default sets of fonts for which the micro-typographic features will be enabled (see table 2) only contain those encodings for which configurations exist. Therefore, if you are using any other encoding (e. g., T2A, LGR etc.), microtype will not apply to these fonts. You have to define and activate a new font set including the encoding(s) you are using (for details, see section 4). For protrusion at least, you would also have to create settings for the fonts in question (see section 5.1). It goes without saying that contributions for these encodings are more than welcome.

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

You might want to disable protrusion in the Table of Contents. In unfortunate situations, enabled protrusion might internally alter the line length in the TOC and similar lists in such a way that an excess leader dot will fit in. The solution is to temporarily disable protrusion for the TOC:

```
\microtypesetup{protrusion=false}
\tableofcontents
\microtypesetup{protrusion=true}
```

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

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

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

Compatibility and interaction with other packages: The `microtype` package is supposed to work happily together with all other \LaTeX packages (except for `pdfcprot`). However, life isn't perfect, so problems are to be expected. Currently, I am only aware of the following issues concerning the loading order:

- If you want to use 8-bit characters in the configuration, you have to load the `inputenc` package first. Unicode input is also supported (when loading `inputenc` with the `utf8` or the `utf8x` option). When using multiple input encodings in a document, 8-bit characters in the settings will only work reliably if you specify the `inputenc` key.
- When loading the package with the `babel` option, you must load the `babel` package before `microtype`.

Possible error messages and how to get rid of them:

- ! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.
This error message will occur if you are trying to employ font expansion while creating DVI output. Remember, that *automatic* font expansion only works when running $\text{pdf}\text{\TeX}$ in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your \TeX system.
- ! pdfTeX error (font expansion): auto expansion is only possible with scalable fonts.
Automatic font expansion has been improved in $\text{pdf}\text{\TeX}$ 1.40, in that it now not only works with Type 1 fonts but also with TrueType, OpenType and even non-embedded fonts. The above error message indicates either that you are trying to apply expansion to a bitmap (pk) font, which is still not possible, or that the font isn't found at all, e. g., because of missing map entries.
- Warning: pdflatex: font ptmr8r cannot be expanded (not an included Type1 font)
and the PDF viewer complains about a missing font, e. g., Adobe Reader thusly:
Could not find a font in the Resources dictionary - using Helvetica instead.
With $\text{pdf}\text{\TeX}$ versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your \TeX system is not set up to embed (or 'download') the base PostScript fonts (e. g., Times, Helvetica, Courier). In most \TeX distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to true.
- Warning: pdflatex (file ecrm1000+20): Font ecrm1000+20 at 1200 not found
Furthermore, $\text{pdf}\text{\TeX}$ versions older than 1.40 require Type 1 fonts for automatic font expansion. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. With older $\text{pdf}\text{\TeX}$ versions, this is only possible if you manually create expanded instances of the fonts.
- ! Font Tl/cmr/m/n/10=ecrm1000 at 10.0pt not loaded: Not enough room left.
Memory parameter 'font_mem_size' too small.

- ! TeX capacity exceeded, sorry [maximum internal font number (font_max)=2000].
Memory parameter 'font_max' too small.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf_mem_size)=65536].
Memory parameter 'pdf_mem_size' too small (pdfTeX versions older than 1.30).
When applying micro-typographic enhancement to a large document with a lot of fonts, pdfTeX may be running out of some kind of memory. It can be increased by setting the respective parameter to a larger value. For web2c-based systems, e. g., T_EXLive, change the settings in texmf.cnf, for MiK_TE_X, in the file miktex.ini (2.4 or older) resp. pdf_latex.ini (2.5 or newer).
- pdfTeX warning (font expansion): font should be expanded before its first use
This warning will occur with pdfTeX versions older than 1.40.4, if tracking *and* expansion is applied to a font. It is harmless and can be ignored.

10 Contributions

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

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

11 Acknowledgments

This package would be pointless if *Hàn Thế Thành* hadn't created the pdfTeX programme in the first place, which introduced the micro-typographic extensions and made them available to the T_EX world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004](#) and [Thành 2008](#). I also thank him and the rest of the pdfTeX team for refuting the idea that T_EX is dead, and for fixing the bugs I find.

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

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

Additionally, the following people have reported bugs, made suggestions or helped otherwise (in chronological order): *Tom Kink, Herb Schulz, Michael Hoppe, Gary L. Gray, Georg Verweyen, Christoph Bier, Peter Muthesius, Bernard Gaulle †, Adam Kucharczyk, Mark Rossi, Stephan Hennig, Michael Zedler, Herbert Voß, Ralf Stubner, Holger Uhr, Peter Dyballa, Morten Høgholm, Steven Bath, Daniel Flipo, Michalis Miatidis, Sven Naumann, Ross Hetherington, Geoff Vallis, Steven E. Harris, Karl Berry, Peter Meier, Nathan Rosenblum and Wolfram Schaalo.*

12 References

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

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

Hàn Thế Thành, ‘Font-specific issues in pdf \TeX ’, in: *TUGboat*, vol. 29(2008), no. 1: ‘EuroBacho \TeX 2007 Proceedings’, pp. 36–41. (Online – for TUG members – at <https://www.tug.org/members/TUGboat/tb29-1/tb91thanh-fonts.pdf>)

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, Martin Schröder, *The pdf \TeX user manual*, 25 January 2007. (Available from CTAN at </systems/pdftex/>; latest version at <http://sarovar.org/projects/pdftex/>)

Karl Berry, *Fontname. Filenames for \TeX fonts*, September 2005. (Available from CTAN at </info/fontname/fontname.pdf>)

\LaTeX 3 Project Team, *\LaTeX 2 ϵ font selection*, 27 November 2005. (Available from CTAN at </macros/latex/doc/fntguide.pdf>)

Carsten Schurig, Tobias Schlemmer, *The pdfcp $\text{\textit{prot}}$.sty package*, 10 June 2005. (Available from CTAN at [/macros/latex/contrib/pdfcp \$\text{\textit{prot}}\$ /](/macros/latex/contrib/pdfcp\textit{prot}/))

Melchior Franz, *The soul package*, 17 November 2003. (Available from CTAN at </macros/latex/contrib/soul/>). See also Heiko Oberdiek’s extension of this package, *soulut $\text{\textit{f8}}$* , which adds Unicode support. (Available from CTAN at </macros/latex/contrib/oberdiek/>)

13 Short history

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

2.3 (2007/12/23)

- New key ‘outer kerning’ for `\SetTracking` to customise outer kerning [section 5.3]
- Adjust protrusion settings for tracking even if protrusion is not enabled
- New option ‘verbose=silent’ to turn all warnings into mere messages [section 3.5]
- The `letterspace` package also works with `eplain` or `minil $\text{\textit{tx}}$` [section 7]

2.2 (2007/07/14)

- Improvements to tracking/letterspacing: retain kerning (with pdfTeX 1.40.4); automatically adjust protrusion settings
- New key ‘no ligatures’ for `\SetTracking` to disable selected or all ligatures (with pdfTeX 1.40.4) [section 5.3]
- New keys ‘spacing’ and ‘outer spacing’ for `\SetTracking` to customise interword spacing [section 5.3]
- Possibility to expand a font with different parameters (with pdfTeX 1.40.4) [section 5.2]
- New optional argument for `\DisableLigatures` to disable selected ligatures only [section 8]
- New command `\DeclareMicrotypeVariants` to specify variant suffixes [section 5.7]
- New command `\textmicrotypecontext` as a wrapper for `\microtypecontext` [section 6]
- Protrusion settings for Bitstream Letter Gothic
- Basic support for LuaTeX (if generated with the `lua` option)

2.1 (2007/01/21)

- New command `\slig` to protect ligatures in letterspaced text [section 7]

2.0 (2007/01/14)

- Support for the new extensions of pdfTeX version 1.40: tracking/letterspacing, adjustment of interword spacing (glue), and additional kerning (new commands `\SetTracking`, `\SetExtraSpacing`, `\SetExtraKerning`; new options ‘tracking’, ‘spacing’, ‘kerning’) [sections 5.3, 5.4, 5.5]
- New commands `\textls` and `\lstyle` for letterspacing, new option ‘letterspace’ [sections 3.4, 7]
- New option ‘babel’ for automatic micro-typographic adjustment to the selected language [sections 3.5, 6]
- New font sets: ‘smallcaps’, ‘footnotesize’, ‘scriptsize’ [section 4; table 2]
- New package ‘letterspace’ providing the commands for robust and hyphenatable letterspacing [section 7]

1.9e (2006/07/28)

- New key ‘inputenc’ to specify the lists’ input encodings [section 5]
- Protrusion settings for Euler math fonts

1.9d (2006/05/05)

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

1.9c (2006/02/02)

- Protrusion settings for URW Garamond

1.9a (2005/12/05)

- Defer setup until the end of the preamble; consequently, no need to change font

defaults before loading `microtype`, or to put it the other way round, `microtype` may now be loaded at any time

- Inside the preamble, `\microtypesetup` accepts all package options [section 3.6]
- Protrusion settings for T5 encoded Charter

1.9 (2005/10/28)

- New command `\DisableLigatures` to disable ligatures of fonts (requires pdf \TeX version 1.30 or later) [section 8]
- New command `\microtypecontext` to change the configuration context; new key ‘context’ for the configuration commands [section 6]
- New key ‘font’ to add single fonts to the font sets [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)

1.8 (2005/06/23)

- New command `\DeclareMicrotypeSetDefault` to declare the default font sets [section 4]
- New option ‘config’ to load a different configuration file [section 3.5]
- New option ‘unit’ to measure protrusion factors relative to a dimension instead of the character width [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
- When using the `ledmac` package, character protrusion will work for the first time ever (requires pdf \TeX version 1.30 or later)

1.7 (2005/03/23)

- Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings [sections 4, 5]
- New command `\LoadMicrotypeFile` to load a font configuration file manually [section 5.7]
- Hook `\Microtype@Hook` for font package authors [section 14.4.3]
- New option ‘verbose=errors’ to turn all warnings into errors
- Warning when running in draft mode

1.6 (2005/01/24)

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

1.5 (2004/12/15)

- When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled [section 3.1]

- New option ‘selected’ to enable selected expansion, default: false [sections 3.3, 5.2]
- New default for expansion option ‘step’: 4 ($\min(\text{stretch}, \text{shrink})/5$) [section 3.3]
- Protrusion settings for Bitstream Charter

1.4 (2004/11/12)

- Set up fonts independently from \LaTeX font loading
- New option: ‘final’ [section 3.5]

1.2 (2004/10/03)

- New font sets: ‘allmath’ and ‘basicmath’ [section 4; table 2]
- Protrusion settings for Computer Modern Roman math symbols
- Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond

1.1 (2004/09/21)

- Protrusion settings for Adobe Minion, contributed by Harald Harders
- New command: `\DeclareCharacterInheritance` [section 5.6]
- Characters may also be specified as octal or hexadecimal numbers [section 5]

1.0 (2004/09/11)

- First CTAN release

14 Implementation

The docstrip modules in this file are:

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

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

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

`lua`: Code for LuaTeX (by default disabled).

`plain`: Code for `eplain`, `miniltx` (letterspace only).

`debug`: Code for additional output in the log file.

Used for – surprise! – debugging purposes.

`config`: Surrounds all configuration modules.

`cfg-t`: Surrounds (Latin) text configurations.

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

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

`blg`: Settings for Bitstream Letter Gothic (`mt-blg.cfg`).

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

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

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

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

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

Contributed by *Harald Harders*.

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

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

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

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

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

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

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

`cfg-e`: Surrounds Euro symbol configurations.

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

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

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

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

And now for something completely different.

¹ `<package|letterspace>`

14.1 Preliminaries

`\MT@MT` This is us.

```
2 \def\MT@MT
3 <package> {microtype}
4 <letterspace> {letterspace}
```

`\MT@fix@catcode` 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.

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

```
5 \let\MT@restore@catcodes\@empty
6 \def\MT@fix@catcode#1#2{%
7   \edef\MT@restore@catcodes{%
8     \MT@restore@catcodes
9     \catcode#1 \the\catcode#1\relax
10  }%
11  \catcode#1 #2\relax
12 }
13 <package>\MT@fix@catcode{17}{14}% ^^Q (comment)
14 \MT@fix@catcode{24} {9}% ^^X (ignore)
15 <package>\MT@fix@catcode{33}{12}% !
16 <package>\MT@fix@catcode{34}{12}% "
17 \MT@fix@catcode{36} {3}% $ (math shift)
18 \MT@fix@catcode{39}{12}% '
19 \MT@fix@catcode{42}{12}% *
20 \MT@fix@catcode{43}{12}% +
21 \MT@fix@catcode{44}{12}% ,
22 \MT@fix@catcode{45}{12}% -
23 \MT@fix@catcode{58}{12}% :
24 \MT@fix@catcode{60}{12}% <
25 \MT@fix@catcode{61}{12}% =
26 \MT@fix@catcode{62}{12}% >
27 <package>\MT@fix@catcode{63}{12}% ?
28 \MT@fix@catcode{94} {7}% ^ (superscript)
29 \MT@fix@catcode{96}{12}% ~
30 <package>\MT@fix@catcode{124}{12}% |
```

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

```
31 <*package>
32 \newcommand*\DeclareMicrotypeSet[3] [] {}
33 \newcommand*\UseMicrotypeSet[2] [] {}
34 \newcommand*\DeclareMicrotypeSetDefault[2] [] {}
35 \newcommand*\SetProtrusion[3] [] {}
36 \newcommand*\SetExpansion[3] [] {}
37 \newcommand*\SetTracking[3] [] {}
38 \newcommand*\SetExtraKerning[3] [] {}
39 \newcommand*\SetExtraSpacing[3] [] {}
40 \newcommand*\DisableLigatures[2] [] {}
41 \newcommand*\DeclareCharacterInheritance[3] [] {}
42 \newcommand*\DeclareMicrotypeVariants[1] {}
43 \newcommand*\DeclareMicrotypeAlias[2] {}
44 \newcommand*\LoadMicrotypeFile[1] {}
45 \newcommand*\DeclareMicrotypeBabelHook[2] {}
46 \newcommand*\microtypesetup[1] {}
47 \newcommand*\microtypecontext[1] {}
48 \newcommand*\textmicrotypecontext[2] {#2}
49 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
50 </package>
```

```

51 \newcommand*\lststyle{}
52 \newcommand\textls[2][]{\textls[2]{}}
53 \def\textls#1#{}
54 \newcommand*\lslig[1]{#1}
55 *package
56 }

```

These commands also have a starred version.

```

57 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
58 \def\DeclareMicrotypeVariants#1#{\@gobble}

```

Set declarations are only allowed in the preamble (resp. the main configuration file). The configuration commands, on the other hand, must be allowed in the document, too, since they may be called inside font configuration files, which, in principle, may be loaded at any time.

```

59 \@onlypreamble\DeclareMicrotypeSet
60 \@onlypreamble\UseMicrotypeSet
61 \@onlypreamble\DeclareMicrotypeSetDefault
62 \@onlypreamble\DisableLigatures
63 \@onlypreamble\DeclareMicrotypeVariants
64 \@onlypreamble\DeclareMicrotypeBabelHook

```

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

```

65 \def\MT@old@cmd#1#2{%
66   \newcommand*#1{\MT@warning{%
67     \string#1 is deprecated. Please use\MessageBreak
68     \string#2 instead}%
69     \let #1#2#2}}
70 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
71 \MT@old@cmd\DeclareMicroTypeSet \DeclareMicrotypeSet
72 \MT@old@cmd\UseMicroTypeSet \UseMicrotypeSet
73 \MT@old@cmd\LoadMicroTypeFile \LoadMicrotypeFile
74 /package

```

`\MT@warning` Communicate.

```

\MT@warning@nl 75 \def\MT@warning{\PackageWarning\MT@MT}
\MT@info 76 \def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
\MT@info@nl 77 *package
\MT@vinfo 78 \def\MT@info{\PackageInfo\MT@MT}
\MT@vinfo 79 \def\MT@info@nl#1{\MT@info{#1\@gobble}}
\MT@error 80 \let\MT@vinfo@gobble
\MT@warn@err 81 \def\MT@error{\PackageError\MT@MT}
82 \def\MT@warn@err#1{\MT@error{#1}{%
83   This error message appears because you loaded the \MT@MT'\MessageBreak
84   package with the option `verbose=errors'. Consult the documentation\MessageBreak
85   in \MT@MT.pdf to find out what went wrong.}}

```

14.1.1 Debugging

`\tracingmicrotype` Cases for `\tracingmicrotype`:

```

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

```

```

86 <debug>
87 \MT@warning@n1{This is the debug version}
88 \newcount\tracingmicrotype
89 \tracingmicrotype=2
90 \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
91 \def\MT@info@n1#1{\PackageInfo\MT@MT{#1\@gobble}\MT@addto@annot{#1}}
92 \let\MT@vinfo\MT@info@n1
93 \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
94 \def\MT@warning@n1#1{\PackageWarning\MT@MT{#1\@gobble}\MT@addto@annot{Warning: #1}}
95 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
96 \def\MT@dinfo@n1#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@n1{#2}\fi}

```

\tracingmicrotypeinpdf

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

Cases for \tracingmicrotypeinpdf:

- 1: show new fonts
- 2: + show known fonts

```
97 \newcount\tracingmicrotypeinpdf
```

Let's see how it works . . .

```
\tracingmicrotypeinpdf=2
```

\MT@pdf@annot

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

\MT@addto@annot

\ifMT@inannot

```

98 \newif\ifMT@inannot \MT@inannottrue
99 \let\MT@pdf@annot\empty
100 \def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
101   {\def\MessageBreak{^J\@spaces}%
102    \MT@xadd\MT@pdf@annot{\pdfescapestring{#1^J}}}\fi\fi}

```

\iftracingmicrotypeinpdfall

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

```
103 \newif\iftracingmicrotypeinpdfall
```

\MT@show@pdf@annot

A red caret is shown for fonts which are actually set up by *Microtype*, a green one marks fonts that we have already seen. The /Caret annotation requires a viewer for PDF version 1.5 (you could use /Text if you're using an older PDF viewer).

```

104 \def\MT@show@pdf@annot#1{%
105   \ifnum\tracingmicrotypeinpdf<#1 \else
106     \iftracingmicrotypeinpdfall\leavevmode\fi
107     \pdfannot height 4pt width 4pt depth 2pt {%
108       /Subtype/Caret
109       /T(\expandafter\string\font@name)
110       \ifcase#1\or
111         /Subj(New font)/C[1 0 0]
112       \else
113         /Subj(Known font)/C[0 1 0]
114       \fi
115       /Contents(\MT@pdf@annot)
116     }%
117     \iftracingmicrotypeinpdfall\kern1pt \fi
118     \global\MT@inannotfalse
119   \fi
120 }
121 </debug>

```

122 `</package>`

14.1.2 Requirements

`\MT@plain` The letterspace package works with:

0: miniltx

1: eplain

2: L^AT_EX

```

123 <*plain>
124 \def\MT@plain{2}
125 \ifx\documentclass\@undefined
126 \def\MT@plain{1}
127 \def\hmode@bgroup{\leavevmode\bgroup}
128 \let\@typeset@protect\relax
129 \ifx\eplain\@undefined
130 \def\MT@plain{0}
131 \def\PackageWarning#1#2{%
132 \begingroup
133 \newlinechar=10 %
134 \def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces\@spaces}%
135 \immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}%
136 \endgroup
137 }
138 \def\on@line{ on input line \the\inputlineno}
139 \def\@spaces{\space\space\space\space}
140 \fi
141 \fi

```

`\MT@requires@latex` Better use groups than plain ifs.

```

142 \def\MT@requires@latex#1{%
143 \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
144 }
145 </plain>

```

`\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, we have to distinguish seven cases for pdf_T_EX:

0: not running pdf_T_EX

1: pdf_T_EX (< 0.14f)

2: + micro-typographic extensions (0.14f,g)

3: + protrusion relative to 1em (≥ 0.14h)

4: + automatic font expansion; protrusion no longer has to be set up first; default `\efcode = 1000` (≥ 1.20)

5: + `\(left,right)marginkern`; `\pdfnoligatures`; `\pdfstrcmp`; `\pdfescapestring` (≥ 1.30)

6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`¹¹; `\pdftracingfonts`; always e-_T_EX (≥ 1.40)

11 This command was actually introduced in 1.30, but failed on strings longer than 1023 bytes.

7: + \letterspacefont doesn't disable ligatures and kerns; \pdfcopyfont ($\geq 1.40.4$)

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

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

```
147 \ifx\normalpdftexversion\@undefined \else
148   \let\pdftexversion\normalpdftexversion
149   \let\pdftexrevision\normalpdftexrevision
150   \let\pdfoutput\normalpdfoutput
151 \fi
```

Old packages might have let \pdftexversion to \relax.

```
152 \ifx\pdftexversion\@undefined \else
153   \ifx\pdftexversion\relax \else
154     <debug>\MT@info@nl{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
155     \def\MT@pdftex@no{7}
156     <package>
157     \ifnum\pdftexversion = 140
158       \ifnum\pdftexrevision < 4
159         \def\MT@pdftex@no{6}
160       \fi
161     \else
162     </package>
163     \ifnum\pdftexversion < 140
164       \def\MT@pdftex@no{5}
165     <package>
166     \ifnum\pdftexversion < 130
167       \def\MT@pdftex@no{4}
168     \ifnum\pdftexversion < 120
169       \def\MT@pdftex@no{3}
170     \ifnum\pdftexversion = 14
171       \ifnum\expandafter`\pdftexrevision < `h
172         \def\MT@pdftex@no{2}
173       \ifnum\expandafter`\pdftexrevision < `f
174         \def\MT@pdftex@no{1}
175       \fi
176     \fi
177   \else
178     \ifnum\pdftexversion < 14
179       \def\MT@pdftex@no{1}
180     \fi
181   \fi
182 \fi
183 \fi
184 \fi
185 </package>
186 \fi
187 \fi
188 \fi
189 <debug>\MT@info@nl{0}{pdftex no.: \MT@pdftex@no}
```

\MT@clear@options If we are not using pdfT_EX or in case it is too old, we disable everything and exit.

```
190 \def\MT@clear@options{%
191   <plain> \MT@requires@latex1{%
192     \AtEndOfPackage{\let\@unprocessedoptions\relax}%
193     \let\CurrentOption\empty
194   <plain> } \relax
195 }
196 \ifnum\MT@pdftex@no <
```

```

197 <package> 2
198 <letterspace> 6
199 \MT@warning@nl{You
200 \ifcase\MT@pdfTeX@no
201 don't seem to be using pdfTeX.\MessageBreak
202 ~\MT@MT' only works with pdfTeX.\MessageBreak
203 Try running 'pdflatex' instead of 'latex'%
204 \else
205 are using a pdfTeX version older than
206 <package> 0.14%
207 <letterspace> 1.40%
208 .\MessageBreak
209 ~\MT@MT' does not work with this version.\MessageBreak
210 Please install a newer version of pdfTeX%
211 \fi
212 }
213 \MT@clear@options\MT@restore@catcodes
214 \endinput\fi

```

Still there? Then we can begin: We need the `keyval` package, including the ‘new’ `\KV@sp@def` implementation.

```

215 \RequirePackage{keyval}[1997/11/10]
216 <*package>

```

```

\MT@toks    We need a token register.
217 \newtoks\MT@toks
\ifMT@if@    A scratch if.
218 \newif\ifMT@if@

```

14.1.3 Declarations

```

\ifMT@protrusion    These are the global switches ...
\ifMT@expansion 219 \newif\ifMT@protrusion
\ifMT@auto 220 \newif\ifMT@expansion
\ifMT@selected 221 \newif\ifMT@auto
222 \newif\ifMT@selected
\ifMT@noligatures 223 \newif\ifMT@noligatures
\ifMT@draft 224 \newif\ifMT@draft
\ifMT@spacing 225 \newif\ifMT@spacing
\ifMT@kerning 226 \newif\ifMT@kerning
\ifMT@tracking 227 \newif\ifMT@tracking
228 \newif\ifMT@babel
\MT@MT@babel ... and numbers.
\MT@ex@level 229 \let\MT@pr@level\tw@
\MT@pr@factor 230 \let\MT@ex@level\tw@
231 \let\MT@pr@factor\@m
\MT@ex@factor 232 \let\MT@ex@factor\@m
\MT@sp@factor 233 \let\MT@sp@factor\@m
\MT@kn@factor 234 \let\MT@kn@factor\@m
\MT@pr@unit    Default unit for protrusion settings is character width, for spacing space, for kerning
\MT@sp@unit    (and tracking) 1em.
\MT@kn@unit 235 \let\MT@pr@unit\@empty
236 \let\MT@sp@unit\@mone
237 \def\MT@kn@unit{1em}
\MT@stretch    Expansion settings.
\MT@shrink
\MT@step 238 \let\MT@stretch\@mone

```

```

239 \let\MT@shrink \m@ne
240 \let\MT@step \m@ne

\MT@pr@min      Minimum and maximum values allowed by pdfTeX.
\MT@pr@max 241 \def\MT@pr@min{-\@m}
\MT@ex@min 242 \let\MT@pr@max\@m
243 \let\MT@ex@min\z@
\MT@ex@max 244 \let\MT@ex@max\@m
\MT@sp@min 245 \def\MT@sp@min{-\@m}
\MT@sp@max 246 \let\MT@sp@max\@m
247 \def\MT@kn@min{-\@m}
\MT@kn@min 248 \let\MT@kn@max\@m
\MT@kn@max 249 /package
250 \def\MT@tr@min{-\@m}
\MT@tr@min 251 \let\MT@tr@max\@m
\MT@tr@max 252 *package

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

\MT@stretch@default Default values for expansion.
\MT@shrink@default 254 \def\MT@stretch@default{20 }
\MT@step@default 255 \def\MT@shrink@default{20 }
256 \def\MT@step@default{4 }

\MT@letterspace Default value for letterspacing (in thousandths of 1em).
\MT@letterspace@default 257 /package
258 \let\MT@letterspace\m@ne
259 \def\MT@letterspace@default{100}
260 *package

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

```

14.1.4 Auxiliary macros

```

\MT@maybe@etex For definitions that depend on e-TeX features.
262 \ifcase 0%
263   \ifx\TeXversion\@undefined 1\else
264     \ifx\TeXversion\relax 1\else
265       \ifcase\TeXversion 1\fi
266     \fi
267   \fi
268 \else
269   \catcode`\^^Q=9 \catcode`\^^X=14
270 \fi
271 debug\MT@info@nl{0}{this is
272 debug^^Q not
273 debug etex}

\MT@requires@pdftex For definitions that depend on a particular pdfTeX version.
274 \def\MT@requires@pdftex#1{%
275   \ifnum\MT@pdftex@no<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
276 }
277 debug\MT@requires@pdftex6{\pdftracingfonts=1}\relax

\MT@requires@luatex For definitions that depend on LuaTeX.
278 *lua
279 \def\MT@requires@luatex{%
280   \ifx\directlua\@undefined\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi

```



```

281 }
282 <debug>\MT@info@n10{this is \MT@requires@luatex{}{not }luatex}

\MT@lua      Communicate with lua.
283 \def\MT@lua{\directlua0}
284 </lua>
285 </package>

Since LuaTeX is still heavily experimental, we don't support it by default. Also, at
the time of this writing, LuaTeX ignores \letterspacefont.

286 <!*lua|letterspace>
287 \ifx\directlua\undefined\else
288 <!letterspace> \MT@error
289 <letterspace> \MT@warning@n1
290   {\MT@MT'
291 <!letterspace> only works with luatex if you generate%
292 <letterspace> doesn't currently work with luatex.%
293   \MessageBreak
294 <!letterspace> the package with the `lua' option%
295 <letterspace> Bye bye%
296   }
297 <!letterspace> {}
298 <letterspace> \MT@clear@options\MT@restore@catcodes\expandafter\endinput
299 \fi
300 </!*lua|letterspace>

\MT@glet      The forgotten primitive.
301 \def\MT@glet{\global\let}

\MT@exp@cs    Commands to create command sequences. Those that are going to be defined
\MT@exp@gcs    globally should be created inside a group so that the save stack won't explode.
302 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
303 <*package>
304 \def\MT@exp@gcs#1#2{\begingroup\expandafter\endgroup\expandafter#1\csname#2\endcsname}

\MT@def@n     This is \@namedef and global.
\MT@gdef@n    305 \def\MT@def@n{\MT@exp@cs\def}
306 \def\MT@gdef@n{\MT@exp@gcs\gdef}

\MT@edef@n    Its expanding versions.
\MT@xdef@n    307 </package>
308 \def\MT@edef@n{\MT@exp@cs\edef}
309 <*package>
310 \def\MT@xdef@n{\MT@exp@gcs\xdef}

\MT@let@nc    \let a \csname sequence to a command.
\MT@glet@nc    311 \def\MT@let@nc{\MT@exp@cs\let}
312 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}

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

\MT@let@nn    \let a \csname sequence to a \csname sequence.
\MT@glet@nn    314 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}
315 \def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}

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

\MT@exp@one@n Expand the second token once and enclose it in braces.
317 </package>

```

```

318 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}
\MT@exp@two@c      Expand the next two tokens after <#1> once.
319 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
320 <*\package>
\MT@exp@two@n      Expand the next two tokens after <#1> once and enclose them in braces.
321 \def\MT@exp@two@n#1#2#3{%
322   \expandafter\expandafter\expandafter
323   #1\expandafter\expandafter\expandafter
324   {\expandafter#2\expandafter}\expandafter{#3}}
You do not wonder why \MT@exp@one@c doesn't exist, do you?
\MT@ifdefined@c@T  Wrapper for testing whether command resp. \csname sequence is defined. If we
\MT@ifdefined@c@TF are running e-TeX, we will use its primitives \ifdefined and \ifcsname, which
\MT@ifdefined@n@T  decreases memory use substantially.
\MT@ifdefined@n@TF 325 \def\MT@ifdefined@c@T#1{%
326   ^^X \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
327   ^^Q \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
328 }
329 </package>
330 \def\MT@ifdefined@c@TF#1{%
331   ^^X \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
332   <package>^^Q \ifx#1\@undefined
333   <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
334 }
335 \def\MT@ifdefined@n@T#1{%
336   ^^X \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
337   <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
338   <package>^^Q \expandafter\@gobble\else\expandafter\@firstofone\fi
339 }
340 <*\package>
341 \def\MT@ifdefined@n@TF#1{%
342   ^^X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
343   ^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
344   ^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
345 }
\MT@detokenize@n  Translate a macro into a token list. With e-TeX, we can use \detokenize (and
\MT@detokenize@c  \expandafter\string to get rid of the trailing space). The e-TeX version requires
some more fiddling (and the \string isn't perfect, of course).
346 \def\MT@detokenize@n#1{%
347   ^^X \detokenize\expandafter{%
348     \string#1%
349   }%
350 }
351 \def\MT@detokenize@c#1{%
352   ^^X \detokenize\expandafter\expandafter\expandafter{\expandafter\string#1}%
353   ^^Q \MT@exp@two@c\zap@space\strip@prefix\meaning#1 \empty
354 }
\MT@ifempty      Test whether argument is empty.
355 </package>
356 \begingroup
357 \catcode`\%=12
358 \catcode`\&=14
359 \gdef\MT@ifempty#1{%
360   \if %#1%
361     \expandafter\@firstoftwo
362   \else
363     \expandafter\@secondoftwo

```

```

364 \fi
365 }
366 \endgroup
367 <package>

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

368 \MT@requires@pdftex6{
369 <lua>
370 \MT@requires@luatex{
371   \def\MT@ifint#1{%
372     \csname \MT@lua{
373       if string.find("\luaescapestring{#1}", "^-[0-9]+ *$")
374       then tex.write("@firstoftwo")
375       else tex.write("@secondoftwo")
376     end}%
377   \endcsname
378 }
379 }{
380 </lua>
381 </package>
382   \def\MT@ifint#1{%
383     \ifcase\pdfmatch{^-[0-9]+ *$}{#1}\relax
384     \expandafter\@secondoftwo
385   \else
386     \expandafter\@firstoftwo
387   \fi
388 }
389 <package>
390 <lua> }
391 }{
392   \def\MT@ifint#1{%
393     \if!\ifnum9<1#1!\else?\fi
394     \expandafter\@firstoftwo
395   \else
396     \expandafter\@secondoftwo
397   \fi
398 }
399 }

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

400 \MT@requires@pdftex6{
401 <lua>
402 \MT@requires@luatex{
403   \def\MT@ifdimen#1{%
404     \csname \MT@lua{
405       if (string.find("\luaescapestring{#1}", "^-[0-9]+(\@percentchar a*) *$") or
406         string.find("\luaescapestring{#1}", "^-[0-9]*[.][0-9]+(\@percentchar a*) *$"))
407       then tex.write("@firstoftwo")
408       else tex.write("@secondoftwo")
409     end}%
410   \endcsname
411 }
412 }{
413 </lua>
414   \def\MT@ifdimen#1{%
415     \ifcase\pdfmatch{^([0-9]+([.][0-9]+)?|[.][0-9]+)%
416       (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
417     \expandafter\@secondoftwo
418   \else

```

```

419     \expandafter\@firstoftwo
420     \fi
421   }
422   <lua> }
423 }{
424   \def\MT@ifdimen#1{%
425     \setbox\z@=\hbox{%
426       \MT@count=1#1\relax
427       \ifnum\MT@count=\@ne
428         \aftergroup\@secondoftwo
429       \else
430         \aftergroup\@firstoftwo
431       \fi
432     }%
433   }
434 }

\MT@ifdim    Test floating point numbers.

435 \def\MT@ifdim#1#2#3{%
436   \ifdim #1\p@ #2 #3\p@
437     \expandafter\@firstoftwo
438   \else
439     \expandafter\@secondoftwo
440   \fi
441 }

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

442 \MT@requires@pdftex5{
443   <lua>
444   \MT@requires@luatex{
445     \def\MT@ifstreq#1#2{%
446       \csname \MT@lua{
447         if "\luaescapestring{#1}" == "\luaescapestring{#2}"
448           then tex.write("@firstoftwo")
449           else tex.write("@secondoftwo")
450         end}%
451       \endcsname
452     }
453   }{
454     </lua>
455     \def\MT@ifstreq#1#2{%
456       \ifcase\pdfstrcmp{#1}{#2}\relax
457         \expandafter\@firstoftwo
458       \else
459         \expandafter\@secondoftwo
460       \fi
461     }
462     <lua> }
463   }{
464     \def\MT@ifstreq#1#2{%
465       \edef\MT@res@a{#1}%
466       \edef\MT@res@b{#2}%
467       \ifx\MT@res@a\MT@res@b
468         \expandafter\@firstoftwo
469       \else
470         \expandafter\@secondoftwo
471       \fi
472     }
473   }

\MT@xadd    Add item to a list.

474 \def\MT@xadd#1#2{%

```

```

475 \ifx#1\relax
476 \xdef#1{#2}%
477 \else
478 \xdef#1{#1#2}%
479 \fi
480 }

\MT@xaddb      Add item to the beginning.
481 \def\MT@xaddb#1#2{%
482 \ifx#1\relax
483 \xdef#1{#2}%
484 \else
485 \xdef#1{#2#1}%
486 \fi
487 }

\MT@map@clist@n      Run <#2> on all elements of the comma list <#1>. This and the following is modelled
\MT@map@clist@c      after LATEX3 commands.
\MT@map@clist@      488 </package>
\MT@clist@function  489 \def\MT@map@clist@n#1#2{%
490 \ifx\@empty#1\else
491 \def\MT@clist@function##1{#2}%
492 \MT@map@clist@#1,\@nil,\@nnil
493 \fi
494 }
495 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
496 \def\MT@map@clist@#1,{%
497 \ifx\@nil#1%
498 \expandafter\MT@clist@break
499 \fi
500 \MT@clist@function{#1}%
501 \MT@map@clist@
502 }
503 \let\MT@clist@function\@gobble
504 \def\MT@clist@break#1\@nnil{}
505 <*package>

\MT@map@tlist@n      Execute <#2> on all elements of the token list <#1>. \MT@tlist@break can be used
\MT@map@tlist@c      to jump out of the loop.
\MT@map@tlist@      506 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break      507 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
508 \def\MT@map@tlist@#1#2{%
509 \ifx\@nnil#2\else
510 #1{#2}%
511 \expandafter\MT@map@tlist@
512 \expandafter#1%
513 \fi
514 }
515 \def\MT@tlist@break#1\@nnil{\fi}

\ifMT@inlist@      Test whether item <#1> is in comma list <#2>. Using \pdfmatch would be slower.
\MT@in@clist      516 \newif\ifMT@inlist@
517 \def\MT@in@clist#1#2{%
518 \def\MT@res@a#1,#1,##2##3\@nnil{%
519 \ifx##2\@empty
520 \MT@inlist@false
521 \else
522 \MT@inlist@true
523 \fi
524 }%
525 \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
526 }

```

<code>\MT@rem@from@clist</code>	Remove item $\langle \#1 \rangle$ from comma list $\langle \#2 \rangle$. This is basically <code>\@removeelement</code> from <code>ltxcntrl.dtx</code> . Using <code>\pdfmatch</code> and <code>\pdflastmatch</code> here would be really slow!
	<pre> 527 \def\MT@rem@from@clist#1#2{% 528 \def\MT@res@a##1,#1,##2\MT@res@a{##1,##2\MT@res@b}% 529 \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}% 530 \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}% 531 } </pre>
<code>\MT@in@tlist</code>	Test whether item is in token list. Since this isn't too elegant, I thought that at least
<code>\MT@in@tlist@</code>	here, <code>\pdfmatch</code> would be more efficient – however, it turned out to be even slower than this solution.
	<pre> 532 \def\MT@in@tlist#1#2{% 533 \MT@inlist@false 534 \def\MT@res@a{#1}% 535 \MT@map@tlist@c#2\MT@in@tlist@ 536 } 537 \def\MT@in@tlist@#1{% 538 \edef\MT@res@b{#1}% 539 \ifx\MT@res@a\MT@res@b 540 \MT@inlist@true 541 \expandafter\MT@tlist@break 542 \fi 543 } </pre>
<code>\MT@in@rlist</code>	Test whether size <code>\MT@size</code> is in a list of ranges. Store the name of the list in
<code>\MT@in@rlist@</code>	<code>\MT@size@name</code>
<code>\MT@in@rlist@@</code>	<pre> 544 \def\MT@in@rlist#1{% 545 \MT@inlist@false 546 \MT@map@tlist@c#1\MT@in@rlist@ 547 } 548 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1} 549 \def\MT@in@rlist@@#1#2#3{% 550 \MT@ifdim{#2}=\m@ne{% 551 \MT@ifdim{#1}=\MT@size 552 \MT@inlist@true 553 \relax 554 }{% 555 \MT@ifdim\MT@size<{#1}\relax{% 556 \MT@ifdim\MT@size<{#2}% 557 \MT@inlist@true 558 \relax 559 }% 560 }% 561 \ifMT@inlist@ 562 \def\MT@size@name{#3}% 563 \expandafter\MT@tlist@break 564 \fi 565 } </pre>
<code>\MT@loop</code>	This is the same as L ^A T _E X's <code>\loop</code> , which we mustn't use, since this could confuse an
<code>\MT@iterate</code>	outer <code>\loop</code> in the document.
<code>\MT@repeat</code>	<pre> 566 <i>/package</i> 567 \def\MT@loop#1\MT@repeat{% 568 \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}% 569 \MT@iterate \let\MT@iterate\relax 570 } 571 \let\MT@repeat\fi </pre>
<code>\MT@while@num</code>	Execute $\langle \#3 \rangle$ from $\langle \#1 \rangle$ up to (excluding) $\langle \#2 \rangle$ (much faster than L ^A T _E X's <code>\@whilenum</code>).
	<pre> 572 \def\MT@while@num#1#2#3{% 573 \@tempcnta#1\relax </pre>

```

574 \MT@loop #3%
575 \advance\@tempcnta \@ne
576 \ifnum\@tempcnta < #2\MT@repeat
577 }

\MT@do@font      Execute <#1> 256 times.
578 \def\MT@do@font{\MT@while@num\z@\cc@lvi}
579 (*package)

\MT@count        Increment macro <#1> by one. Saves using up too many counters. The e-TeX way is
\MT@increment     slightly faster.
580 \newcount\MT@count
581 \def\MT@increment#1{%
582 ^^X \edef#1{\number\numexpr #1 + 1\relax}%
583 ^^Q \MT@count=#1\relax
584 ^^Q \advance\MT@count \@ne
585 ^^Q \edef#1{\number\MT@count}%
586 }

\MT@scale        Multiply and divide a counter. If we are using e-TeX, we will use its \numexpr
                  primitive. This has the advantage that it is less likely to run into arithmetic overflow.
                  The result of the division will be rounded instead of truncated. Therefore, we'll get
                  a different (more accurate) result in about half of the cases.
587 \def\MT@scale#1#2#3{%
588 ^^Q \multiply #1 #2\relax
589 \ifnum #3 = \z@
590 ^^X #1=\numexpr #1 * #2\relax
591 \else
592 ^^X #1=\numexpr #1 * #2 / #3\relax
593 ^^Q \divide #1 #3\relax
594 \fi
595 }

\MT@abbr@pr      Some abbreviations. Thus, we can have short command names but full-length log
\MT@abbr@ex      output.
\MT@abbr@pr@c    596 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c    597 \def\MT@abbr@ex{expansion}
\MT@abbr@pr@inh  598 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@ex@inh  599 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh  600 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@nl      601 \def\MT@abbr@ex@inh{expansion inheritance}
\MT@abbr@nl      602 \def\MT@abbr@nl{noligatures}
\MT@abbr@sp      603 \def\MT@abbr@sp{spacing}
\MT@abbr@sp@c    604 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@sp@inh  605 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@kn      606 \def\MT@abbr@kn{kerling}
\MT@abbr@kn@c    607 \def\MT@abbr@kn@c{kerling codes}
\MT@abbr@kn@inh  608 \def\MT@abbr@kn@inh{kerling inheritance}
\MT@abbr@tr      609 \def\MT@abbr@tr{tracking}
\MT@abbr@tr@c    610 \def\MT@abbr@tr@c{tracking amount}

\MT@abbr@tr      \MT@abbr@tr
\MT@rbba@protrusion  These we also need the other way round.
\MT@rbba@expansion  \MT@abbr@tr@c
\MT@rbba@expansion  611 \def\MT@rbba@protrusion{pr}
\MT@rbba@spacing   612 \def\MT@rbba@expansion{ex}
\MT@rbba@kerling   613 \def\MT@rbba@spacing{sp}
\MT@rbba@tracking  614 \def\MT@rbba@kerling{kn}
\MT@rbba@tracking  615 \def\MT@rbba@tracking{tr}

\MT@features      We can work on these lists to save some guards in the dtx file.
\MT@features@long 616 \def\MT@features{pr,ex,sp,kn,tr}

```

```

617 \def\MT@features@long{protrusion,expansion,spacing,kerning,tracking}
\MT@is@feature    Whenever an optional argument accepts a list of features, we can use this com-
                  mand to check whether a feature exists in order to prevent a rather confusing
                  ‘Missing \endcsname inserted’ error message. The feature (long form) must be in
                  \@tempa, the type of list to ignore in (#1), then comes the action.
618 \def\MT@is@feature#1{%
619   \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
620   \ifMT@inlist@
621     \expandafter\@firstofone
622   \else
623     \MT@error{\@tempa' is not an available micro-typographic\MessageBreak
624       feature. Ignoring #1}{Available features are: \MT@features@long'.}%
625     \expandafter\@gobble
626   \fi
627 }

```

14.1.5 Compatibility

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

- \pickup@font
- \do@subst@correction
- \add@accent (all in section 14.2.9)
- \showlyphens (in section 14.4.5)

The wordcount package redefines the font-switching commands, which will break microtype. Since microtype doesn’t have an effect on the number of words in the document anyway, we will simply disable ourselves.

```

628 \@ifl@aded{tex}{wordcount}{%
629   \MT@warning@nl{Detected the `wordcount' utility.\MessageBreak
630     Disabling \MT@MT', since it wouldn't work}%
631   \MT@clear@options\MT@restore@catcodes\endinput}\relax
\MT@setup@    The setup is deferred until the end of the preamble. This has a couple of advantages:
              \microtypesetup can be used to change options later on in the preamble, and fonts
              don't have to be set up before microtype.
632 \</package>
633 \<plain>\MT@requires@latex1{
634 \let\MT@setup@\@empty
\MT@addto@setup    We use our private hook to have better control over the timing. This will also work
                  with eplain, but not with miniltx alone.
635 \def\MT@addto@setup{\g@addto@macro\MT@setup@
                  It will be executed at the end of the preamble, and emptied (the combine class calls
                  it repeatedly).
636 \AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}
                  Don't hesitate with miniltx.
637 \<plain>}{\let\MT@addto@setup\@firstofone}
\MT@with@package@T    We almost never do anything if a package is not loaded.
638 \def\MT@with@package@T#1{\@ifpackage@loaded{#1}\@firstofone\@gobble}
639 \<*package>

```


`\MT@with@babel@and@T` L^AT_EX's `\ifpackagewith` ignores the class options.

```

640 \def\MT@with@babel@and@T#1{%
641   \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
642     \expandtwoargs\MT@in@clist{#1}
643     {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
644     \ifMT@inlist@expandafter@gobble\fi
645   }@gobble
646 }

```

Don't load `letterspace`.

```

647 \MT@let@nc{ver@letterspace.sty}\@empty

```

`\MT@ledmac@setup` The `ledmac` package first saves each paragraph in a box, from which it then splits
`\MT@led@unhbox@line` off the lines one by one. This will destroy character protrusion. (There aren't any
`\MT@led@kern` problems with the `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 pdf_T_EX 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 pdf_T_EX 1.21b (aka. 1.30.0).

```

648 \MT@requires@pdftex5{
649   \def\MT@ledmac@setup{%
650     \ifMT@protrusion
651       \MT@ifdefined@c@TF\l@dunhbox@line{%
652         \MT@info@nl{Patching ledmac to enable character protrusion}%
653         \newdimen\MT@led@kern
654         \let\MT@led@unhbox@line\l@dunhbox@line
655         \renewcommand*{\l@dunhbox@line}[1]{%
656           \ifhbox##1%
657             \MT@led@kern=\rightmarginkern##1%
658             \kern\leftmarginkern##1%
659             \MT@led@unhbox@line##1%
660             \kern\MT@led@kern
661           \fi
662         }%
663       }{%
664         \MT@warning@nl{%
665           Character protrusion in paragraphs with line\MessageBreak
666           numbering will only work if you update ledmac}%
667       }%
668     \fi
669   }
670 }{
671   \def\MT@ledmac@setup{%
672     \ifMT@protrusion
673       \MT@warning@nl{%
674         The pdftex version you are using does not allow\MessageBreak
675         character protrusion in paragraphs with line\MessageBreak
676         numbering by the 'ledmac' package.\MessageBreak
677         Upgrade pdftex to version 1.30 or later}%
678       \fi
679     }
680 }

```

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

```

681 \def\MT@restore@p@h{\chardef\%~\% \chardef\#~\# }

```

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

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

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

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

Spanish (and Galician and Mexican) `babel` modify `\%`, storing the original meaning in `\percentsign`.

```
683 \MT@if@false
684 \MT@with@babel@and@T{spanish} \MT@if@true
685 \MT@with@babel@and@T{galician} \MT@if@true
686 \MT@with@babel@and@T{mexican} \MT@if@true
687 \ifMT@if@MT@ifdefined@c@T\percentsign{\let\%\percentsign}\fi
```

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.)

```
688 \MT@with@package@T{csquotes}{%
689 \ifpackage@later{csquotes}{2005/05/11}\@disablequotes\relax}%
```

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

```
690 \MT@if@false
691 \MT@with@package@T{hyperref} \MT@if@true
692 \MT@with@package@T{tex4ht} \MT@if@true
693 \ifMT@if@MT@restore@p@h\fi
694 }
```

Check again at the end of the preamble.

```
695 
```

Our competitor, the `pdfcpot` package, must not be tolerated!

```
698 \MT@with@package@T{pdfcpot}{%
699 \MT@error{Detected the 'pdfcpot' package!\MessageBreak
700 '\MT@MT' and 'pdfcpot' may not be used together}{%
701 The 'pdfcpot' package provides an interface to character protrusion.\MessageBreak
702 So does the '\MT@MT' package. Using both packages at the same\MessageBreak
703 time will almost certainly lead to undesired results. Have your choice!}%
704 }
705 \MT@with@package@T{ledmac}\MT@ledmac@setup
```

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

```
706 \let\MT@setupfont@hook\@empty
707 \MT@if@false
708 \MT@with@babel@and@T{spanish} \MT@if@true
709 \MT@with@babel@and@T{galician} \MT@if@true
710 \MT@with@babel@and@T{mexican} \MT@if@true
711 \ifMT@if@
712 \g@addto@macro\MT@setupfont@hook{%
713 \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
714 \fi
715 \MT@with@package@T{csquotes}{%
716 \ifpackage@later{csquotes}{2005/05/11}{%
717 \g@addto@macro\MT@setupfont@hook\@disablequotes
```

```

718 }{%
719   \MT@warning@n!{%
720     Should you receive warnings about unknown slot\MessageBreak
721     numbers, try upgrading the `csquotes' package}%
722 }%
723 }

```

We disable microtype's additions inside hyperref's `\pdfstringdef`, which redefines lots of commands. hyperref doesn't work with plain \TeX , so in that case we don't bother.

```

724 \MT@if@false
725 </package>
726 <plain> \MT@requires@latex2{
727   \MT@with@package@T{hyperref}{%
728     \pdfstringdefDisableCommands{%
729 <*package>
730       \let\pickup@font\MT@orig@pickupfont
731       \let\textmicrotypecontext\@secondoftwo
732       \let\microtypecontext\@gobble
733 </package>
734       \def\lsstyle{\pdfstringdefWarn\lsstyle}%
735       \def\textls#1#{\pdfstringdefWarn\textls}%
736     }%
737 <package> \MT@if@true
738   }
739 <plain> }\relax
740 <*package>
741 \MT@with@package@T{tex4ht}\MT@if@true
742 \ifMT@if@lg@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi

```

The listings package makes numbers and letters active,

```

743 \MT@with@package@T{listings}{%
744   \g@addto@macro\MT@cfg@catcodes{%
745     \MT@while@num{"30}{ "3A}{\catcode\@tempcnta 12\relax}%
746     \MT@while@num{"41}{ "5B}{\catcode\@tempcnta 11\relax}%
747     \MT@while@num{"61}{ "7B}{\catcode\@tempcnta 11\relax}%
748   }%

```

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

```

749   \g@addto@macro\MT@setupfont@hook{%
750     \catcode`\@z@

```

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

```

751   \let\lst@ProcessLetter\empty
752 }%
753 }

```

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

```

754 </package>
755 <plain> \MT@requires@latex2{
756   \MT@with@package@T{soul}{%
757     \soulregister\lsstyle 0%
758     \soulregister\textls 1%
759   }

```

Under plain \TeX , `soul` doesn't register itself the \LaTeX way, hence we have to use a different test in this case.

```

760 <*plain>
761   {\ifx\Soul@\undefined\else
762     \soulregister\lsstyle 0%
763     \soulregister\textls 1%
764     \fi}
765 </plain>
766 <*package>

```

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

```

767 \MT@with@package@T{pinyin}{%
768   \let\MT@orig@py@macron\py@macron
769   \ifpackage@later{pinyin}{2006/10/17}{% 4.7.0
770     \def\py@macron#1#2{%
771       \let\pickup@font\MT@orig@pickupfont
772       \MT@orig@py@macron{#1}{#2}%
773       \let\pickup@font\MT@pickupfont}%
774     }{%
775       \def\py@macron#1{%
776         \let\pickup@font\MT@orig@pickupfont
777         \MT@orig@py@macron{#1}%
778         \let\pickup@font\MT@pickupfont}%
779       }%
780     }
781 </package>
782 }
783 <*package>

```

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

```

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

```

14.2 Font Setup

`\MT@setupfont` Setting up a font entails checking for each feature whether it should be applied to the current font (`\MT@font`). But first, We might have to disable stuff when used together with adventurous packages.

```

785 \def\MT@setupfont{\MT@setupfont@hook}

```

This will use a copy of the font (allowing for expansion parameter variation and the use of more than one set of protrusion factors for a font within one paragraph).

```

786 \MT@requires@pdftex7
787 {\g@addto@macro\MT@setupfont\MT@copy@font}\relax

```

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

```

788 \g@addto@macro\MT@setupfont{%
789   \MT@exp@two@c\MT@split@name\string\MT@font/\@nil

```

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

```

790   \MT@exp@one@n\MT@find@file\MT@family
791   \ifx\MT@familyalias\@empty \else
792     \MT@exp@one@n\MT@find@file\MT@familyalias\fi

```

We have to make sure that `\cf@encoding` expands to the correct value (for later, in `\MT@get@slot`), which isn't the case when `\selectfont` chooses a new encoding

(this would be done a second later in `\selectfont`, anyway – three lines, to be exact). (I think, I do not need this anymore – however, I’m too afraid to remove it. ... Oops, I did it. Let’s see whether anybody complains.)

```
793 % \ifx\fontencoding\cfontencoding\else\@enc@update\fi
794 }
```

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

```
795 \MT@requires@pdftex6
796 {\g@addto@macro\MT@setupfont\MT@tracking}\relax
797 \g@addto@macro\MT@setupfont{%
798   \MT@check@font
799   \ifMT@inlist@
800     <debug>\MT@showpdfannot2%
801   \else
802     \MT@vinfo{Setting up font `~\MT@font'\on@line}%
```

Now we can begin setting up the font for all features that the current pdfTeX provides. The following commands are `\let` to `\relax` if the respective feature is disabled via package options.

For versions older than 1.20, protrusion has to be set up first, beginning with 1.20, the order doesn’t matter.

```
803 \MT@protrusion
804 \MT@expansion
805 }
```

Interword spacing and kerning (pdfTeX 1.40).

```
806 \MT@requires@pdftex6
807 {\g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}}\relax
```

Disable ligatures (pdfTeX 1.30).

```
808 \MT@requires@pdftex5
809 {\g@addto@macro\MT@setupfont\MT@noligatures}\relax
810 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
811 <debug>\MT@showpdfannot1%
```

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

```
812 \MT@register@font
813 \fi
814 }
```

`\MT@copy@font` The new `\pdfcopyfont` command allows to expand a font with different parameters, or to use more than one set of protrusion factors for a given font within one paragraph. It will be used when we find a context for `\SetProtrusion` or `\SetExpansion` in the preamble, or when the package has been loaded with the `copyfonts` option.

```
815 \let\MT@copy@font\relax
816 \MT@requires@pdftex7{
817 \def\MT@copy@font@{%
```

`\MT@font@copy` For every new protrusion and expansion contexts, we create a new copy.

```
818 \xdef\MT@font@copy{\csname\MT@font/\MT@pr@context/\MT@ex@context\endcsname}%
```

`\MT@font@orig` pdfTeX doesn’t allow to copy a font that has already been copied and expanded/letterspaced. Hence, we have to get the original.

```
819 \expandafter\ifx\MT@font@copy\relax
820   \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
821 \expandafter\ifx\MT@font@orig\relax
822   \MT@exp@two@c\MT@glet\MT@font@orig\font@name
```

```

823 \else
824 \MT@exp@two@c\let\font@name\MT@font@orig
825 \fi
826 \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
827 <debug>\MT@dinfol{creating new copy: \MT@font@copy}%

```

Since it's a new font, we have to remove it from the context lists.

```

828 \MT@map@clist@c\MT@active@features{%
829 \MT@exp@cs@ifx{MT@\@nameuse{MT@abbr@#1}}\relax\else
830 \def\@tempa{#1}%
831 \MT@exp@cs\MT@map@tlist@c{MT@#1@doc@contexts}\MT@rem@from@list
832 \fi
833 }%
834 \fi
835 \MT@exp@two@c\let\MT@font\MT@font@copy

```

We only need the font identifier for letterspacing.

```

836 \let\font@name\MT@font@copy

```

But we have to properly substitute the font after we're done.

```

837 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
838 }

```

\MT@rem@from@list

```

839 \def\MT@rem@from@list#1{%
840 \MT@exp@cs@ifx{MT@\@tempa @#1font@list}\relax\else
841 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
842 \MT@font \csname MT@\@tempa @#1font@list\endcsname
843 \fi
844 }
845 }\relax

```

Here's the promised dirty trick for users of older pdfTeX versions, which works around the problem that the use of the same font with different expansion parameters is prohibited. 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 can load a minimally larger font for the paragraph in question. E. g., for a document typeset in 10pt:

```

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

```

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.

\MT@split@name
\MT@encoding
\MT@family
\MT@series
\MT@shape
\MT@size

Split up the font name (`(/6)` may be a protrusion/expansion context and/or a letterspacing amount).

```

846 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
847   \def\MT@encoding{#1}%
848   \def\MT@family   {#2}%
849   \def\MT@series   {#3}%
850   \def\MT@shape    {#4}%
851   \def\MT@size     {#5}%

```

\MT@familyalias **Alias family?**

```

852   \MT@ifdefined@n@TF{MT@\MT@family @alias}%
853   {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
854   {\let\MT@familyalias\@empty}%
855 }

```

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

\MT@maybe@do 856 \newif\ifMT@do
857 \def\MT@maybe@do#1{%
 (but only if the feature isn't globally set to false)
858 \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).
859 \MT@dotrue
860 \edef\@tempa{\csname MT@#1\setname\endcsname}%
861 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
862 \MT@ifdefined@n@TF{MT@checklist@#1}%
863 {\csname MT@checklist@#1\endcsname}%
864 {\MT@checklist@{#1}}%
865 {#1}%
866 }%
867 \else
868 \MT@dofalse
869 \fi
870 \ifMT@do
 \MT@feat stores the current feature.
871 \def\MT@feat{#1}%
872 \csname MT@set@#1\codes\endcsname
873 \else
874 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
875 \fi
876 }

\MT@dinfo@list

```

877 <debug> \def\MT@dinfo@list#1#2#3{\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#1}:
878 <debug> #2 \ifx\#3\list empty\else \@nameuse{MT@#2}' #3 list\fi}%
879 <debug>}

```

\MT@checklist@ The generic test (**<#1>** is the axis, **<#2>** the feature, **\@tempa** contains the set name).

```

880 \def\MT@checklist@#1#2{%
881 <!debug> \MT@ifdefined@n@T
882 <debug> \MT@ifdefined@n@TF
883   {MT@#2list@#1@\@tempa}{%

```

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

```

884   \expandafter\MT@exp@one@n\expandafter\MT@in@clist
885   \csname MT@#1\expandafter\endcsname
886   \csname MT@#2list@#1@\@tempa\endcsname
887   \ifMT@inlist@

```

```

888 <debug>\MT@info@list{#2}{#1}{in}%
889     \MT@dotrue
890     \else
891 <debug>\MT@info@list{#2}{#1}{not in}%
892     \MT@dofalse
893     \expandafter\MT@clist@break
894     \fi
895 }%

```

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

```

896 <debug> {\MT@info@list{#2}{#1}{}}%
897 }

```

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

```

898 \def\MT@checklist@family#1{%
899 <!debug> \MT@ifdefined@n@T
900 <debug> \MT@ifdefined@n@TF
901     {MT@#1list@family@ \@tempa}{%
902     \MT@exp@two@n\MT@in@clist
903     \MT@family{\csname MT@#1list@family@ \@tempa\endcsname}%
904     \ifMT@inlist@
905 <debug>\MT@info@list{#1}{family}{in}%
906     \MT@dotrue
907     \else
908 <debug>\MT@info@list{#1}{family}{not in}%
909     \MT@dofalse
910     \ifx\MT@familyalias\@empty \else
911     \MT@exp@two@n\MT@in@clist
912     \MT@familyalias{\csname MT@#1list@family@ \@tempa\endcsname}%
913     \ifMT@inlist@
914 <debug> \MT@info@list{#1}{familyalias}{in}%
915     \MT@dotrue
916 <debug>\else\MT@info@list{#1}{familyalias}{not in}%
917     \fi
918     \fi
919     \fi
920     \ifMT@do \else
921     \expandafter\MT@clist@break
922     \fi
923 }%
924 <debug> {\MT@info@list{#1}{family}{}}%
925 }

```

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

```

926 \def\MT@checklist@size#1{%
927 <!debug> \MT@ifdefined@n@T
928 <debug> \MT@ifdefined@n@TF
929     {MT@#1list@size@ \@tempa}{%
930     \MT@exp@cs\MT@in@rlist{MT@#1list@size@ \@tempa}%
931     \ifMT@inlist@
932 <debug>\MT@info@list{#1}{size}{in}%
933     \MT@dotrue
934     \else
935 <debug>\MT@info@list{#1}{size}{not in}%
936     \MT@dofalse
937     \expandafter\MT@clist@break
938     \fi
939 }%
940 <debug> {\MT@info@list{#1}{size}{}}%
941 }

```



```
942 \def\MT@checkboxlist@font#1{%
943 \!debug \MT@ifdefined@n@T
944 debug \MT@ifdefined@n@TF
945 {MT@#1list@font\@tempa}{%
```

```

946 \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
947 \expandafter\MT@exp@one@n\expandafter\MT@in@clst\expandafter
948 \@tempb \csname MT@#1listofont@\@tempa\endcsname
949 \ifMT@inlist@
950 (debug)\MT@dinfo@list{#1}{font}{in}%
951 \expandafter\MT@cclst@break
952 \else
953 (debug)\MT@dinfo@list{#1}{font}{not in}%
954 \MT@dofalse
955 \fi
956 }%
957 (debug) {\MT@dinfo@list{#1}{font}{}}%
958 }

```

\MT@protrusion Set up for protrusion?

`\MT@set@pr@codes` This macro is called by `\MT@setupfont`, and does all the work for setting up a font for protrusion.

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

Get the name of the inheritance list and parse it.

Set an input encoding?

Load additional lists?

Load the main list.

```
\MT@gobble@settings
```

<code>\MT@get@font@dimen@six</code>	If <code>\fontdimen 6</code> is zero, character protrusion won't work, and we can skip the
<code>\MT@dimen@six</code>	settings (for example, the <code>dsfont</code> fonts don't specify this dimension; this is probably a bug).

```

974 \def\MT@get@font@dimen@six{%
975   \ifnum\fontdimen6\MT@font=\z@
976     \MT@warning@n1{%
977       Font '\MT@font' does not specify its\MessageBreak
978       \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
979       \@nameuse{MT@abbr@\MT@feat} will not work with this font}%
980     \expandafter\MT@gobble@settings
981   \else
982     \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
983   \fi
984 }

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

985 \def\MT@set@all@pr#1#2{%
986   debug\MT@info@n1{3}{-- lp/rp: setting all to #1/#2}%
987   \let\MT@temp\@empty
988   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lpcode\MT@font\@tempcnta=#1\relax}}%
989   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rpcode\MT@font\@tempcnta=#2\relax}}%
990   \MT@do@font\MT@temp
991 }

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

992 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
993 \let\MT@reset@pr@codes\relax

\MT@the@pr@code    If the font is letterspaced, we have to add half the letterspacing amount to the
\MT@the@pr@code@tr  margin kerns. This will be activated in \MT@set@tr@codes.

994 \def\MT@the@pr@code{\@tempcntb}
995 \MT@requires@pdftex6{
996   \def\MT@the@pr@code@tr{%
997     \numexpr\@tempcntb+\MT@letterspace@/2\relax
998   }
999 } \relax

\MT@set@codes    Split up the values and set the codes.

1000 \def\MT@set@codes#1,{%
1001   \ifx\relax#1\@empty\else
1002     \MT@split@codes #1==\relax
1003     \expandafter\MT@set@codes
1004   \fi
1005 }

\MT@split@codes    The keyval package would remove spaces here, which we needn't do since
                    \SetProtrusion ignores spaces in the protrusion list anyway. \MT@get@char@unit
                    may mean different things.

1006 \def\MT@split@codes#1=#2=#3\relax{%
1007   \def\@tempa{#1}%
1008   \ifx\@tempa\@empty \else
1009     \MT@get@slot
1010     \ifnum\MT@char > \m@ne
1011       \MT@get@char@unit
1012       \csname MT@\MT@feat @split@val\endcsname#2\relax
1013     \fi
1014   \fi
1015 }

\MT@pr@split@val

1016 \def\MT@pr@split@val#1,#2\relax{%
1017   \def\@tempb{#1}%

```

```

1018 \MT@ifempty\@tempb\relax{%
1019   \MT@scale@to@em
1020   \lpcode\MT@font\MT@char=\MT@the@pr@code
1021 (debug)\MT@info{n1}{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char: [#1]}%
1022 }%
1023 \def\@tempb{#2}%
1024 \MT@ifempty\@tempb\relax{%
1025   \MT@scale@to@em
1026   \rpcode\MT@font\MT@char=\MT@the@pr@code
1027 (debug)\MT@info{n1}{4}{;;; rp (\MT@char): \number\rpcode\MT@font\MT@char: [#2]}%
1028 }%

```

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

```

1029 \MT@ifdefined@c@T\MT@pr@inh@name{%
1030   \MT@ifdefined@nT{\MT@inh@\MT@pr@inh@name @\MT@char @}{%
1031     \MT@exp@cs\MT@map@tlist@c
1032     {\MT@inh@\MT@pr@inh@name @\MT@char @}%
1033     \MT@set@pr@heirs
1034   }%
1035 }%
1036 }

```

`\MT@scale@to@em`

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

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

```

1037 \MT@requires@pdftex3{
1038   \def\MT@scale@to@em{%
1039     \@tempcntb=\MT@count\relax

```

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

```

1040   \MT@scale\@tempcntb \@tempb \MT@dimen@six
1041   \ifnum\@tempcntb=\z@ \else
1042     \MT@scale@factor
1043   \fi
1044 }

```

`\MT@get@charwd`

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

```

1045 \def\MT@get@charwd{%
1046   ^^X \MT@count=\fontcharwd\MT@font\MT@char\relax
1047   ^^Q \setbox\z@=\hbox{\MT@font \char\MT@char}%
1048   ^^Q \MT@count=\wd\z@
1049   \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1050 }

```

For letterspaced fonts, we have to subtract the letterspacing amount from the characters’ widths. The protrusion amounts will be adjusted in `\MT@set@pr@codes`.

The letterspaced font is already loaded so that `1em = \fontdimen 6`.

```

1051 \MT@requires@pdftex6{
1052   \g@addto@macro\MT@get@charwd{%
1053     \MT@ifdefined@c@T\MT@letterspace@
1054     {\advance\MT@count-\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1055   }
1056 } \relax
1057 }{

```

No adjustment with versions 0.14f and 0.14g.

```

1058 \def\MT@scale@to@em{%
1059   \MT@count=\@tempb\relax
1060   \ifnum\MT@count=z@ \else
1061     \MT@scale@factor
1062   \fi
1063 }

```

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

```

1064 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1065 }

```

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

```

1066 \def\MT@get@font@dimen#1{%
1067   \ifnum\fontdimen#1\MT@font=z@
1068     \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1069       \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1070       You should use a different 'unit' for \MT@curr@list@name}%
1071   \else
1072     \MT@count=\fontdimen#1\MT@font
1073   \fi
1074 }

```

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

```

1075 \def\MT@info@missing@char{%
1076   \MT@info@n1{Character '\the\MT@toks'
1077     ^^X \iffontchar\MT@font\MT@char
1078     has a width of 0pt
1079     ^^X \else is missing\fi
1080     ^^Q \MessageBreak (it's probably missing)
1081     \MessageBreak in font '\MT@font'. \MessageBreak
1082     Ignoring protrusion settings for this character}%
1083 }

```

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

```

1084 \def\MT@scale@factor{%
1085   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1086     \expandafter\MT@scale\expandafter \@tempcntb
1087     \csname MT@\MT@feat @factor@\endcsname \@m
1088   \fi
1089   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
1090     \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1091   \else
1092     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1093       \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1094     \fi
1095   \fi
1096 }

```

`\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.

```

1097 \def\MT@warn@code@too@large#1{%
1098   \@tempcnta=#1\relax
1099   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1100     \expandafter\MT@scale\expandafter\@tempcnta\expandafter
1101     \@m \csname MT@\MT@feat @factor@\endcsname
1102   \fi
1103   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1104   \MT@warning@n1{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
1105     is too large for character\MessageBreak
1106     ~\the\MT@toks' in \MT@curr@list@name.\MessageBreak
1107     Setting it to the maximum of \number\@tempcnta}%
1108   \@tempcntb=#1\relax
1109 }

```

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

```

1110 \def\MT@get@opt{%
1111   \MT@set@listname

```

\MT@pr@factor@ Apply a factor?

```

\MT@sp@factor@ 1112 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 1113   \MT@let@nn{MT@\MT@feat @factor@}
1114   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1115   \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1116     \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1117   }{%
1118     \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1119   }%

```

\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@ 1120 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1121   \MT@let@nn{MT@\MT@feat @unit@}%
1122   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1123   \MT@exp@cs@ifx{MT@\MT@feat @unit@}\@empty
1124   \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1125     relative to character widths}%
1126   \else
1127     \MT@exp@cs@ifx{MT@\MT@feat @unit@}\m@ne
1128     \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1129       relative to width of space}%
1130   \fi
1131   \fi
1132   }{%
1133     \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
1134   }%

```

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

\MT@get@char@unit

```

1135 \let\MT@get@char@unit\relax
1136 \let\MT@get@space@unit\@gobble
1137 \MT@exp@cs@ifx{MT@\MT@feat @unit@}\@empty
1138 \let\MT@get@char@unit\MT@get@charwd
1139 \else
1140 \MT@exp@cs@ifx{MT@\MT@feat @unit@}\m@ne
1141 \let\MT@get@space@unit\MT@get@font@dimen
1142 \else
1143 \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}%
1144 \fi
1145 \fi

```

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

```

1146 \MT@ifdefined@n@T{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @preset}{%
1147 \csname MT@preset@MT@feat\endcsname
1148 \MT@let@nc{MT@reset@MT@feat @codes}\relax
1149 }%
1150 }

```

\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.

```

1151 \def\MT@get@unit#1{%
1152 \expandafter\MT@get@unit@#1 e!\@nil
1153 \ifx\x\@empty\else\let#1\x\fi
1154 \@defaultunits\@tempdima#1 pt\relax\@nnil
1155 \ifdim\@tempdima=\z@
1156 \MT@warning@n1{%
1157 Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
1158 width. Setting factors of list ~\@nameuse{MT@MT@feat @c@name}'\MessageBreak
1159 relative to character widths instead}%
1160 \let#1\@empty
1161 \let\MT@get@char@unit\MT@get@charwd
1162 \else
1163 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
1164 to \the\@tempdima}%
1165 \MT@count=\@tempdima\relax
1166 \fi
1167 }
1168 \def\MT@get@unit@#1e#2#3\@nil{%
1169 \ifx\#3\\\let\x\@empty \else
1170 \if m#2%
1171 \edef\x{#1\fontdimen6\MT@font}%
1172 \else
1173 \if x#2%
1174 \edef\x{#1\fontdimen5\MT@font}%
1175 \fi
1176 \fi
1177 }
1178 }

```

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

```

1179 \def\MT@set@inputenc#1{%

```

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

```

1180 \def\MT@cat{#1}%
1181 \edef\@tempa{MT@MT@feat @#1@csname MT@MT@feat @#1@name\endcsname @inputenc}%
1182 \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1183 }

```

\MT@set@inputenc@ More recent versions of inputenc remember the current encoding, so that we can test whether we really have to load the encoding file.

```

1184 \MT@addto@setup{%
1185 \ifpackageloaded{inputenc}{
1186 \ifpackageolder{inputenc}{2006/02/22}{
1187 \def\MT@set@inputenc@{%
1188 \MT@ifstreq\inputencodingname{csname\@tempa\endcsname}\relax
1189 \MT@load@inputenc
1190 }
1191 }{
1192 \let\MT@set@inputenc@\MT@load@inputenc

```

```

1193     }
1194   }{
1195     \def\MT@set@inputenc{%
1196       \MT@warning@n1{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1197         \MessageBreak package isn't loaded. Ignoring input encoding}%
1198     }
1199   }
1200 }

```

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

```

1201 \def\MT@load@inputenc{%
1202   \MT@cfg@catcodes
1203   <debug>\MT@info@n1{1}{loading input encoding: \@nameuse{\@tempa}}%
1204   \inputencoding{\@nameuse{\@tempa}}%
1205 }

```

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

```

1206 \def\MT@set@pr@heirs#1{%
1207   \lcode\MT@font#1=\lcode\MT@font\MT@char
1208   \rcode\MT@font#1=\rcode\MT@font\MT@char
1209   <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1210   <debug>\MT@info@n1{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
1211   <debug>                                     \number\rcode\MT@font\MT@char}%
1212 }

```

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

```

\MT@preset@pr@ 1213 \def\MT@preset@pr{%
1214   \expandafter\expandafter\expandafter\MT@preset@pr@
1215   \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1216 }
1217 \def\MT@preset@pr@#1,#2\@nil{%
1218   \ifx\MT@pr@unit@\empty
1219     \MT@warn@preset@twidth{pr}%
1220     \let\MT@preset@aux\MT@preset@aux@factor
1221   \else
1222     \def\MT@preset@aux{\MT@preset@aux@space2}%
1223   \fi
1224   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1225   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1226   \MT@set@all@pr@tempa\@tempb
1227 }

```

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

```

\MT@preset@aux@factor 1228 \def\MT@preset@aux@factor#1#2{%
1229   \@tempcntb=#1\relax
\MT@preset@aux@space 1230   \MT@scale@factor
1231   \edef#2{\number\@tempcntb}%
1232 }
1233 \def\MT@preset@aux@space#1#2#3{%
1234   \def\@tempb{#2}%
1235   \MT@get@space@unit#1%
1236   \MT@scale@to@em
1237   \edef#3{\number\@tempcntb}%
1238 }

```

`\MT@warn@preset@twidth`

```

1239 \def\MT@warn@preset@twidth#1{%
1240   \MT@warning@n1{%
1241     Cannot preset characters relative to their widths\MessageBreak
1242     for \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1243     \MessageBreak relative to lem instead}%

```

1244 }

14.2.2 Expansion

`\MT@expansion` Set up for expansion?

1245 `\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).

```
1246 \def\MT@set@ex@codes@{%
1247   \MT@if@list@exists{%
1248     \MT@get@ex@opt
1249     \let\MT@get@char@unit\relax
1250     \MT@reset@ef@codes
1251     \MT@get@inh@list
1252     \MT@set@inputenc{c}%
1253     \MT@load@list\MT@ex@code@name
1254     \MT@set@listname
1255     \MT@let@cn\@tempc{\MT@ex@code@\MT@ex@code@name}%
1256     \expandafter\MT@set@codes\@tempc,\relax,%
1257     \MT@expandfont
1258   }\relax
1259 }
```

`\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.

```
1260 \newif\ifMT@nonselected
1261 \def\MT@set@ex@codes@n{%
1262   \MT@nonselectedtrue
1263   \MT@if@list@exists
1264   \MT@get@ex@opt
1265   {%
1266     \let\MT@stretch@ \MT@stretch
1267     \let\MT@shrink@ \MT@shrink
1268     \let\MT@step@ \MT@step
1269     \let\MT@auto@ \MT@auto
1270     \let\MT@ex@factor@ \MT@ex@factor
1271   }%
1272   \MT@reset@ef@codes
1273   \MT@expandfont
1274   \MT@nonselectedfalse
1275 }
```

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

1276 `\let\MT@set@ex@codes\MT@set@ex@codes@n`

`\MT@expandfont` Expand the font.

```
1277 \def\MT@expandfont{%
1278   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1279 }
```

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

1280 `\def\MT@set@all@ex#1{%`


```

1281 <debug>\MT@info{n1}{3}{-- ex: setting all to \number#1}%
1282 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1283 }
1284 \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.

```

1285 \MT@requires@pdfTeX4{
1286 \def\MT@reset@ef@codes{%
1287 \ifnum\MT@ex@factor@=\@m \else
1288 \MT@reset@ef@codes@
1289 \fi
1290 }
1291 }{
1292 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1293 }

```

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

```

1294 \def\MT@ex@split@val#1\relax{%
1295 \@tempcntb=#1\relax

```

Take an optional factor into account.

```

1296 \ifnum\MT@ex@factor@=\@m \else
1297 \MT@scale\@tempcntb \MT@ex@factor@ \@m
1298 \fi
1299 \ifnum\@tempcntb > \MT@ex@max
1300 \MT@warn@ex@too@large\MT@ex@max
1301 \else
1302 \ifnum\@tempcntb < \MT@ex@min
1303 \MT@warn@ex@too@large\MT@ex@min
1304 \fi
1305 \fi
1306 \efcode\MT@font\MT@char=\@tempcntb
1307 <debug>\MT@info{n1}{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%

```

Heirs, heirs, I love thy heirs.

```

1308 \MT@ifdefined@c@T\MT@ex@inh@name{%
1309 \MT@ifdefined@n@T{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1310 \MT@exp@cs\MT@map@tlist@c{\MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
1311 }%
1312 }%
1313 }

```

\MT@warn@ex@too@large

```

1314 \def\MT@warn@ex@too@large#1{%
1315 \MT@warning{n1}{Expansion factor \number\@tempcntb\space too large for
1316 character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1317 Setting it to the maximum of \number#1}%
1318 \@tempcntb=#1\relax
1319 }

```

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

```

\MT@ex@factor@ 1320 \def\MT@get@ex@opt{%
\MT@stretch@ 1321 \MT@set@listname
1322 \MT@ifdefined@n@TF{\MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1323 \MT@let@cn\MT@ex@factor@{\MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1324 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 1325 }{%
1326 \let\MT@ex@factor@\MT@ex@factor
1327 }%
1328 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1329 \MT@get@ex@opt@{shrink}{Setting shrink limit to \number\MT@shrink@}%
1330 \MT@get@ex@opt@{step}{Setting expansion step to \number\MT@step@}%

```

```

1331 \def\@tempa{autoexpand}%
1332 \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1333 \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @preset}}}%
1334 \MT@preset@ex
1335 \let\MT@reset@ef@codes\relax
1336 }%
1337 }

```

\MT@get@ex@opt@

```

1338 \def\MT@get@ex@opt@#1#2{%
1339 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}}}%
1340 \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
1341 \MT@vinfo{... : #2}%
1342 }}%
1343 \MT@let@nn{MT@#1@}{MT@#1}%
1344 }%
1345 }

```

\MT@set@ex@heirs

```

1346 \def\MT@set@ex@heirs#1{%
1347 \efcode\MT@font#1=\efcode\MT@font\MT@char
1348 <debug>\MT@info@nl{2}{-- heir of \MT@char: #1}%
1349 <debug>\MT@info@nl{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1350 }

```

\MT@preset@ex

```

1351 \def\MT@preset@ex{%
1352 \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
1353 \MT@scale@factor
1354 \MT@set@all@ex@\@tempcntb
1355 }

```

14.2.3 Interword spacing (glue)

\MT@spacing Adjustment of interword spacing?

```

1356 \MT@requires@pdftex6{
1357 \def\MT@spacing{\MT@maybe@do{sp}}

```

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

```

1358 \def\MT@set@sp@codes{%
1359 \MT@if@list@exists{%
1360 \MT@get@font@dimen@six
1361 \MT@get@opt
1362 \MT@reset@sp@codes
1363 \MT@get@inh@list
1364 \MT@set@inputenc{c}%
1365 \MT@load@list\MT@sp@c@name
1366 \MT@set@listname
1367 \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
1368 \expandafter\MT@set@codes\@tempc,\relax,%
1369 }\MT@reset@sp@codes
1370 }

```

\MT@sp@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).

```

1371 \def\MT@sp@split@val#1,#2,#3\relax{%
1372 \def\@tempb{#1}%
1373 \MT@ifempty\@tempb\relax{%
1374 \MT@get@space@unit2%
1375 \MT@scale@to@em

```

```

1376 \knbscode\MT@font\MT@char=\@tempcntb
1377 <debug>\MT@info@n1{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1378 }%
1379 \def\@tempb{#2}%
1380 \MT@ifempty\@tempb\relax{%
1381 \MT@get@space@unit3%
1382 \MT@scale@to@em
1383 \stbscode\MT@font\MT@char=\@tempcntb
1384 <debug>\MT@info@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1385 }%
1386 \def\@tempb{#3}%
1387 \MT@ifempty\@tempb\relax{%
1388 \MT@get@space@unit4%
1389 \MT@scale@to@em
1390 \shbscode\MT@font\MT@char=\@tempcntb
1391 <debug>\MT@info@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1392 }%
1393 \MT@ifdefined@c@T\MT@sp@inh@name{%
1394 \MT@ifdefined@n@T\MT@inh@MT@sp@inh@name @\MT@char @}%
1395 \MT@exp@cs\MT@map@tlist@c\MT@inh@MT@sp@inh@name @\MT@char @\MT@set@sp@heirs
1396 }%
1397 }%
1398 }

```

\MT@set@sp@heirs

```

1399 \def\MT@set@sp@heirs#1{%
1400 \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1401 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1402 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1403 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1404 <debug>\MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1405 <debug> \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1406 }

```

\MT@set@all@sp

```

\MT@reset@sp@codes 1407 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1408 <debug>\MT@info@n1{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1409 \let\MT@temp\@empty
1410 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
1411 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1412 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1413 \MT@do@font\MT@temp
1414 }
1415 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1416 \let\MT@reset@sp@codes\relax

```

\MT@preset@sp

```

\MT@preset@sp@ 1417 \def\MT@preset@sp{%
1418 \expandafter\expandafter\expandafter\MT@preset@sp@
1419 \csname MT@sp@c@MT@sp@c@name @preset\endcsname\@nil
1420 }
1421 \def\MT@preset@sp@#1,#2,#3\@nil{%
1422 \ifx\MT@sp@unit@\@empty
1423 \MT@warn@preset@to@width{sp}%
1424 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
1425 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
1426 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
1427 \else
1428 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
1429 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
1430 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
1431 \fi

```

```

1432 \MT@set@all@sp\@tempa\@tempc\@tempb
1433 }
1434 }\relax

```

14.2.4 Additional kerning

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

```

1435 \MT@requires@pdftex6{
1436 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

1437 \def\MT@set@kn@codes{%
1438 \MT@if@list@exists{%
1439 \MT@get@font@dimen@six
1440 \MT@get@opt
1441 \MT@reset@kn@codes
1442 \MT@get@inh@list
1443 \MT@set@inputenc{c}%
1444 \MT@load@list\MT@kn@cc@name
1445 \MT@set@listname
1446 \MT@let@cn\@tempc{MT@kn@cc@MT@kn@cc@name}%
1447 \expandafter\MT@set@codes\@tempc,\relax,%
1448 }\MT@reset@kn@codes
1449 }

```

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

```

1450 \def\MT@kn@split@val#1,#2\relax{%
1451 \def\@tempb{#1}%
1452 \MT@if@empty\@tempb\relax{%
1453 \MT@get@space@unit2%
1454 \MT@scale@to@em
1455 \knbcode\MT@font\MT@char=\@tempcntb
1456 (debug)\MT@info@n1{4}{;;; knbc (\MT@char): \number\knbcode\MT@font\MT@char: [#1]}%
1457 }%
1458 \def\@tempb{#2}%
1459 \MT@if@empty\@tempb\relax{%
1460 \MT@get@space@unit2%
1461 \MT@scale@to@em
1462 \knacode\MT@font\MT@char=\@tempcntb
1463 (debug)\MT@info@n1{4}{;;; knac (\MT@char): \number\knacode\MT@font\MT@char: [#2]}%
1464 }%
1465 \MT@if@defined@c@T\MT@kn@inh@name{%
1466 \MT@if@defined@n@T{MT@inh@MT@kn@inh@name @\MT@char @}{%
1467 \MT@exp@cs\MT@map@tlist@c{MT@inh@MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1468 }%
1469 }%
1470 }

```

`\MT@set@kn@heirs`

```

1471 \def\MT@set@kn@heirs#1{%
1472 \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1473 \knacode\MT@font#1=\knacode\MT@font\MT@char
1474 (debug)\MT@info@n1{2}{-- heir of \MT@char: #1}%
1475 (debug)\MT@info@n1{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1476 (debug) \number\knacode\MT@font\MT@char}%
1477 }

```

`\MT@set@all@kn`

```

\MT@reset@kn@codes 1478 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1479 (debug)\MT@info@n1{3}{-- knac/knbc: setting all to #1/#2}%
1480 \let\MT@temp\@empty

```

```

1481 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbcode\MT@font\@tempcnta=#1\relax}}%
1482 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knaccode\MT@font\@tempcnta=#2\relax}}%
1483 \MT@do@font\MT@temp
1484 }
1485 \def\MT@reset@kn@codes@{\MT@set@all@kn@z@z@}
1486 \let\MT@reset@kn@codes\relax

\MT@preset@kn
\MT@preset@kn@ 1487 \def\MT@preset@kn{%
1488 \expandafter\expandafter\expandafter\MT@preset@kn@
1489 \csname MT@kn@cc@MT@kn@cc@name @preset\endcsname\@nil
1490 }
1491 \def\MT@preset@kn@#1,#2\@nil{%
1492 \ifx\MT@kn@unit@\@empty
1493 \MT@warn@preset@towidth{kn}%
1494 \let\MT@preset@aux\MT@preset@aux@factor
1495 \else
1496 \def\MT@preset@aux{\MT@preset@aux@space2}%
1497 \fi
1498 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1499 \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1500 \MT@set@all@kn\@tempa\@tempb
1501 }
1502 }\relax

```

14.2.5 Tracking

This only works with pdfTeX 1.40.

```

1503 \MT@requires@pdftex6{

\MT@tracking    We only check whether a font should not be letterspaced at all, not whether we've
\MT@tracking@    already done that (because we have to do it again).

\MT@tr@font@list 1504 \let\MT@tr@font@list\@empty
1505 \def\MT@tracking@{%
1506 \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
1507 \ifMT@inlist\@else
1508 \MT@maybe@do{tr}%
1509 \ifMT@do\@else
1510 \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1511 \fi
1512 \fi
1513 }
1514 /package
1515 \let\MT@tracking
1516 /package \MT@tracking@
1517 /letterspace \relax

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

1518 \def\MT@set@tr@codes{%
1519 *package
1520 \MT@vinfo{Tracking font ~\MT@font'\on@line}%
1521 \MT@get@font@dimen@six
1522 \MT@if@list@exists
1523 \MT@get@tr@opt
1524 \relax
1525 /package
1526 \MT@ifdefined@c@TF\MT@letterspace@\relax{\let\MT@letterspace@\MT@letterspace}%
1527 \ifnum\MT@letterspace@=\z@\@else
1528 /package \MT@vinfo{... Tracking by \number\MT@letterspace@}%

```

Letterspacing only works in PDF mode.

```
1529 \MT@warn@tracking@DVI
```

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

In contrast to `\MT@font`, which may reflect the font characteristics more accurately (taking substitutions into account), `\font@name` is guaranteed to correspond to an actual font identifier.

```
1530 \xdef\MT@lsfont{\csname\expandafter\string\font@name
1531 \number\MT@letterspace@ls\endcsname}%
1532 \expandafter\ifx\MT@lsfont\relax
1533 <debug>\MT@info@n1{1}{... new letterspacing instance}%
```

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

```
1534 \MT@get@ls@basefont
1535 \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@
```

Scale interword spacing (not configurable in letterspace).

```
1536 <package>
1537 \MT@ifdefined@c@TF\MT@tr@ispace
1538 {\let\@tempa\MT@tr@ispace}%
1539 {\edef\@tempa{\MT@letterspace@*,,}}%
1540 \MT@ifdefined@c@TF\MT@tr@ospace
1541 {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1542 {\edef\@tempa{\@tempa,,}}%
1543 \expandafter\MT@tr@set@space\@tempa,%
1544 </package>
1545 <letterspace>
1546 % spacing = {<letterspace amount>*,,}
1547 \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@relax sp
1548 * \fontdimen2\MT@lsfont/1000relax
1549 </letterspace>
```

Adjust outer kerning (microtype only).

```
1550 <package>
1551 \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
1552 \expandafter\MT@tr@set@okern\@tempa,%
```

Disable ligatures (not configurable in letterspace).

```
1553 \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1554 </package>
1555 <letterspace>
1556 % no ligatures = {f}
1557 \tagcode\MT@lsfont`f=\m@ne
1558 </letterspace>
```

Adjust protrusion values now, and maybe later (in `\MT@pr@split@val`).

```
1559 <debug>\MT@info@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%
1560 \MT@do@font{\lcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2relax
1561 \rprcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2relax}%
1562 <package> \let\MT@the@pr@code\MT@the@pr@code@tr
1563 \fi
```

Finally, let the letterspaced font propagate.

```
1564 \aftergroup\MT@set@lsfont
1565 <package> \let\MT@font\MT@lsfont
```

`\MT@set@curr@ls` We need to remember the current letterspacing amount (for `\slig`).

```
\MT@curr@ls 1566 \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
1567 \aftergroup\MT@set@curr@ls
```

Adjust surrounding spacing and kerning.

`\MT@set@curr@os` We get the current outer spacing and adjust it, then, after the end of the current outer group, set the current outer spacing, again, and adjust.

```
1568 <package>
1569   \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
1570   \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1571   \aftergroup\aftergroup\aftergroup\MT@set@curr@os
1572   \MT@tr@outer@
1573 </package>
```

If `\MT@ls@adjust` is empty, it's the starred version of `\textls`. Use scaling to avoid a 'Dimension too large'.

```
1574   \ifx\MT@ls@adjust\@empty
1575 <letterspace>   % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}
1576   \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1577   \MT@ls@outer@k
1578 <letterspace>
1579   \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1580   \aftergroup\aftergroup\aftergroup\MT@set@curr@ok
1581   \aftergroup\aftergroup\aftergroup\MT@ls@outer@k
1582 </letterspace>
```

Otherwise, get the current outer kerning and adjust it, for left and right side (microtype only).

```
1583 <package>
1584   \else
1585   \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
1586   \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1587   \ifdim\MT@outer@kern=\z@ \else \MT@ls@outer@k \fi
1588   \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
1589   \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1590 </package>
1591   \fi
1592 <package>
```

`\MT@set@curr@ok` Carry the outer kerning amount to outside the next group, then set outer spacing (which will set kerning, if no space follows).

```
1593   \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1594   \aftergroup\aftergroup\aftergroup\MT@set@curr@ok
1595   \aftergroup\aftergroup\aftergroup\MT@tr@outer@
1596 </package>
1597   \fi
1598 }
```

`\MT@get@tr@opt` Various settings (only for the microtype version).

```
1599 <package>
1600 \def\MT@get@tr@opt{%
1601   \MT@set@listname
1602   \MT@ifdefined@n@T{MT@tr@cn\MT@tr@cc@MT@tr@cc@name}{%
1603     \MT@let@cn\MT@letterspace{MT@tr@cc@MT@tr@cc@name}%
```

`\MT@tr@unit@` Different unit?

```
1604   \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name @unit}{%
1605     \MT@let@cn\MT@tr@unit@{MT@tr@cc@MT@tr@cc@name @unit}%
1606     \ifdim\MT@tr@unit@=1em
1607       \let\MT@tr@unit@\@undefined
1608     \else
1609       \MT@let@cn\@tempb{MT@tr@cc@MT@tr@cc@name}%
1610       \MT@get@unit\MT@tr@unit@
1611       \let\MT@tr@factor@\@m
```

```

1612      \MT@scale@to@em
1613      \edef\MT@letterspace{\number\@tempcntb}%
1614      \fi
1615    }%
1616  }%

\MT@tr@ispace    Adjust interword spacing.
\MT@tr@ospace 1617 \MT@get@tr@opt@{spacing}    {ispace}%
1618 \MT@get@tr@opt@{outerspace}{ospace}%

\MT@tr@okern    Adjust outer kerning.
1619 \MT@get@tr@opt@{outerkerning}{okern}%

\MT@tr@ligatures    Which ligatures should we disable (empty means all, undefined none)?
1620 \MT@get@tr@opt@{noligatures} {ligatures}%
1621 }

\MT@get@tr@opt@
1622 \def\MT@get@tr@opt@#1#2{%
1623   \MT@ifdefined@n@T{MT@tr@c@MT@tr@c@name @#1}%
1624   {\MT@let@nn{MT@tr@#2}{MT@tr@c@MT@tr@c@name @#1}}%
1625 }
1626 /package

\MT@set@lsfont    Redefine \font@name, which will be called a second later (in \selectfont).
1627 plain\MT@requires@latex2{
1628 \def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}

\lsstyle    Disable the tests whether the font should be letterspaced, then trigger the setup.
1629 \DeclareRobustCommand\lsstyle{%
1630 package \def\MT@feat{tr}%
1631 \let\MT@tracking\MT@set@tr@codes
1632 \selectfont
1633 }

    Now the definitions for the letterspace package with plain TEX.
1634 plain
1635 }{
1636 \def\MT@set@lsfont{\MT@lsfont}
1637 \def\lsstyle{%
1638   \begingroup
1639   \escapechar\m@ne
1640   \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1641   \MT@set@tr@codes
1642   \endgroup
1643 }
1644 \let\textls\@undefined
1645 \let\lslig\@undefined
1646 }
1647 /plain

\lslig    For Fraktur fonts, some ligatures shouldn't be broken up. This command will
\MT@lslig    temporarily select the base font and insert the correct kerning.
1648 \DeclareRobustCommand\lslig[1]{%
1649   {\MT@ifdefined@c@TF\MT@curr@ls{%
1650     \escapechar\m@ne
1651     \MT@get@ls@basefont
1652     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1653     \kern\MT@outer@kern
1654     \font@name #1%
1655     \kern\MT@outer@kern%
1656   }{#1}}%

```


1657 }

`\MT@get@ls@basefont` pdf_T_EX cannot letterspace fonts that already are letterspaced. Therefore, we have to save the base font in `\font name`@base.

The previous solution (checking the macro's meaning with `\pdfmatch`), where we were loading the base font via the `\font` primitive again, could destroy all previously set up micro-typographic features of the font.

```
1658 \def\MT@get@ls@basefont{%
1659   \edef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1660   \expandafter\ifx\MT@ls@basefont\relax
1661     \MT@exp@two@c\MT@glet\MT@ls@basefont\font@name
1662   \else
1663     (debug)\MT@dinfoln{1}{... fixing base font}%
1664     \MT@exp@two@c\let\font@name\MT@ls@basefont
1665   \fi
1666 }
```

`\MT@tr@noligatures` pdf_T_EX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.

```
1667 (*package)
1668 \MT@requires@pdftex7{
1669   \def\MT@tr@noligatures{%
1670     \ifx\MT@tr@ligatures\empty
1671       \MT@noligatures@\MT@lsfont\undefined
1672     \else
1673       \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1674     \fi
1675   }
1676 }{
1677   \def\MT@tr@noligatures{%
1678     \MT@warning@nl{%
1679       Disabling selected ligatures is only possible since\MessageBreak
1680       pdftex 1.40.4. Disabling all ligatures instead}%
1681     \MT@glet\MT@tr@noligatures\relax
1682   }
1683 }
```

`\MT@outer@space` A new skip for outer spacing.

```
1684 \newskip\MT@outer@space
```

`\MT@tr@set@space` Adjust interword spacing (`\fontdimen 2–4`) for inner and outer space. For inner spacing, the font dimensions will be adjusted, the settings for outer spacing will be remembered in a macro.

```
1685 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
1686   (debug)\MT@dinfoln2{... orig. space: \the\fontdimen2\MT@lsfont,
1687   (debug)   \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
1688   (debug)   \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
1689   \let\MT@temp\empty
1690   \MT@tr@set@space@{#1}{#4}{2}\empty
1691   \MT@tr@set@space@{#2}{#5}{3}\plus
1692   \MT@tr@set@space@{#3}{#6}{4}\minus
1693   \MT@glet@c\MT@outer@space\expandafter\string\font@name\MT@temp
1694   (debug)\MT@dinfoln2{... inner space: \the\fontdimen2\MT@lsfont,
1695   (debug)   \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
1696   (debug)\MT@dinfoln2{... outer space: \MT@temp}%
1697 }
```

`\MT@tr@set@space@` If outer spacing settings don't exist, they will be inherited from the inner spacing settings.

```
1698 \def\MT@tr@set@space@#1#2#3#4{%
1699   \MT@ifempty{#2}{%
```

```

1700 \MT@ifempty{#1}{%
1701 \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
1702 }{%
1703 \MT@tr@set@space@@{#1}{#3}{1000}%
1704 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1705 \fontdimen#3\MT@lsfont=\@tempdima
1706 }%
1707 }{%
1708 \MT@tr@set@space@@{#2}{#3}{2000}%
1709 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1710 \MT@ifempty{#1}\relax{%
1711 \MT@tr@set@space@@{#1}{#3}{1000}%
1712 \fontdimen#3\MT@lsfont=\@tempdima
1713 }%
1714 }%
1715 }

```

`\MT@tr@set@space@@` If the value is followed by an asterisk, the fontdimen will be scaled by the respective amount, otherwise the value denotes the desired dimension in the respective unit.

```

1716 \def\MT@tr@set@space@@#1#2#3{%
1717 \MT@test@ast#1*\@nil{%
1718 \MT@ifdefined@cTF\MT@tr@unit@
1719 {\edef\@tempb{#1}\MT@scale@to@em}
1720 {\@tempcntb=#1\relax}%
1721 \@tempdima=\dimexpr \dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
1722 -\fontdimen#2\MT@lsfont\relax

```

For `\fontdimen 2`, we also have to subtract the kerning that letterspacing adds to the sides of the characters (only half if it's for outer spacing).

```

1723 \ifnum#2=\tw@
1724 \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
1725 \fi
1726 \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
1727 }{%
1728 \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
1729 \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
1730 }%
1731 (debug)\MT@info@n13{... : font dimen #2 (#1): \the\@tempdima}%
1732 }

```

`\MT@tr@outer@l` Recall the last skip (must really be an interword space, not just a marker, nor a ‘hard’ space, i. e., one that doesn’t contain stretch or shrink parts).

```

1733 \def\MT@tr@outer@l{%
1734 \ifhmode
1735 \ifdim\lastskip>5sp
1736 \edef\x{\the\lastskip minus 0pt}%
1737 \setbox\z@\hbox{\MT@outer@space=\x}%
1738 \ifdim\wd\z@>\z@
1739 (debug)\MT@info@2{[[[ adjusting pre space: \the\MT@outer@space}%
1740 \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

1741 \let\MT@ls@outer@k\relax
1742 \else

```

The ragged2e package sets `\spaceskip` without glue.

```

1743 \ifdim\lastskip=%
1744 \ifnum\spacefactor<2000
1745 \spaceskip
1746 \else
1747 \ifdim\xspaceskip=\z@
1748 \dimexpr\spaceskip+\fontdimen7\font@name\relax

```

```

1749         \else
1750         \xspaceskip
1751         \fi
1752     \fi
1753 (debug) \MT@edinfo2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
1754         \unskip \hskip\MT@outer@space\relax
1755         \let\MT@ls@outer@k\relax
1756     \fi
1757 \fi
1758 \fi
1759 \fi
1760 }

```

\MT@tr@outer@r The following is borrowed from soul. I've added the cases for italic correction, since tracking may also be triggered by text commands (e.g., \textsc).

```

\MT@tr@outer@next 1761 \def\MT@tr@outer@r{%
1762     \futurelet\MT@tr@outer@next\MT@tr@outer@r@
1763 }
1764 \def\MT@tr@outer@r@{%
1765     \def\MT@temp*#1{\ifhmode\hskip\MT@outer@space
1766 (debug) \MT@edinfo2{[[[ adjusting post spaces (1): \the\MT@outer@space}%
1767     \fi}%
1768     \ifcat\egroup\noexpand\MT@tr@outer@next
1769     \unkern\egroup
1770     \MT@set@curr@ok
1771     \MT@set@curr@os
1772     \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\x=%}
1773     \else

```

If the next token is \maybe@ic (from an enclosing text command), we gobble it, read the next one, feed it to \maybe@ic@ (via \MT@tr@outer@icr) and then call ourselves again.

```

1774     \ifx\maybe@ic\MT@tr@outer@next
1775     \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\x=%}
1776     \else

```

If the next token is \check@icr (from an inner text command), we insert ourselves just before it. This will then call \maybe@ic again the next round (which however will always insert an italic correction, since it doesn't read beyond our group).

```

1777     \ifx\check@icr\MT@tr@outer@next
1778     \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\x=%}
1779     \else
1780     \ifx\@sptoken\MT@tr@outer@next
1781     \def\MT@temp* {\ifhmode\hskip\MT@outer@space
1782 (debug) \MT@edinfo2{[[[ adjusting post spaces (2): \the\MT@outer@space}%
1783     \fi}%
1784     \else
1785     \ifx-\MT@tr@outer@next
1786     \def\MT@temp*~{\nobreak\hskip\MT@outer@space
1787 (debug) \MT@edinfo2{[[[ adjusting post spaces (3): \the\MT@outer@space}%
1788     }%
1789     \else
1790     \ifx\ \MT@tr@outer@next \else
1791     \ifx\space\MT@tr@outer@next \else
1792     \ifx\@xobeysp\MT@tr@outer@next \else

```

If there's no outer spacing, there may be outer kerning.

```

1793     \def\MT@temp*{\ifdim\MT@outer@kern=z@else\MT@ls@outer@k\fi}%
1794     \let\MT@tr@outer@next\relax
1795     \fi\fi\fi\fi\fi\fi\fi\fi
1796     \MT@temp*%

```

```

1797 }

\MT@tr@outer@icr    Helper macros for the italic correction mess.
\MT@tr@outer@icr@ 1798 \def\MT@tr@outer@icr@\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
1799 \def\MT@tr@outer@icr@{%
1800   \let\@let@token= \MT@tr@outer@next
1801   \maybe@ic@
1802 }

    For older pdfTEX versions, throw an error.
1803 {}
1804 \DeclareRobustCommand\lsstyle{%
1805   \MT@error{Letterspacing only works with pdftex version 1.40\MessageBreak
1806     or newer}{Upgrade pdftex, or use the 'soul' package instead.}%
1807   \MT@glet\lsstyle\relax
1808 }
1809 }
1810 </package>

\textls    This command may be used like the other text commands.
1811 \DeclareRobustCommand\textls{%
1812   \hmode\bgroup
1813   \@ifstar{\let\MT@ls@adjust\@empty\MT@textls}%
1814   {\let\MT@ls@adjust\relax \MT@textls}%
1815 }

\MT@textls    The optional argument may be used to change the letterspacing factor.
\MT@letterspace@ 1816 \newcommand\MT@textls[2][{}]{%
1817   \MT@ifempty{#1}%
1818   {\let\MT@letterspace@\@undefined}%
1819   {\KV@esp@def\MT@letterspace@{#1}%
1820     \MT@ls@too@large\MT@letterspace@}%
1821   \lsstyle #2%
1822   \egroup
1823 }

\MT@ls@too@large    Test whether letterspacing amount is too large.
1824 \def\MT@ls@too@large#1{%
1825   \ifnum#1>\MT@tr@max
1826     \MT@warning{Maximum for option 'letterspace' is \number\MT@tr@max}%
1827     \let#1\MT@tr@max
1828   \else
1829     \ifnum#1<\MT@tr@min
1830       \MT@warning{Minimum for option 'letterspace' is \number\MT@tr@min}%
1831       \let#1\MT@tr@min
1832     \fi
1833   \fi
1834 }

\MT@outer@kern    This dimen is used for the starred version of \textls, for \lslig and for adjusted
\MT@tr@set@okern    outer kerning.
1835 \newdimen\MT@outer@kern
1836 <*package>
1837 \def\MT@tr@set@okern#1,#2,{%
1838   \let\MT@temp\@empty
1839   \MT@tr@set@okern@{#1}%
1840   \MT@tr@set@okern@{#2}%
1841   \MT@glet@nc{\MT@outer@kern\expandafter\string\font@name}\MT@temp
1842   <debug>\MT@edinfo@n12{... outer kerning: (#1,#2)
1843   <debug>      = \@nameuse{\MT@outer@kern\expandafter\string\font@name}}}%
1844 }

```

`\MT@tr@set@okern@`

```

1845 \def\MT@tr@set@okern@#1{%
1846   \MT@test@ast#1*\@nil{%
1847     \MT@ifdefined@c@TF\MT@tr@unit@
1848     {\edef\@tempb{#1}\MT@scale@to@em}
1849     {\@tempcntb=#1\relax}%
1850     \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
1851   }%
1852   \MT@ifempty\@tempa{\let\@tempa\@m}\relax
1853   \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
1854   * \fontdimen6\MT@lsfont/2000\relax
1855 }%
1856 \advance\@tempdima -\dimexpr \MT@letterspace@ sp
1857 * \fontdimen6\MT@lsfont/2000\relax
1858 \edef\MT@temp{\MT@temp{\the\@tempdima}}%
1859 }
1860 </package>

```

`\MT@ls@outer@k` Adjust outer kerning.

```

1861 \def\MT@ls@outer@k{\ifhmode\kern\MT@outer@kern\relax\fi}
1862 <*package>

```

14.2.6 Disabling ligatures

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

```

1863 \MT@requires@pdftex5{
1864 \def\MT@noligatures{%
1865   \MT@dotrue
1866   \let\@tempa\MT@nl@setname
1867   \MT@map@c@list@n{font,encoding,family,series,shape,size}%
1868   \MT@ifdefined@c@TF\MT@checklist@##1}%
1869   {\csname MT@checklist@##1\endcsname}%
1870   {\MT@checklist@{##1}}%
1871   {nl}%
1872 }%
1873 \ifMT@do
1874   \MT@noligatures@MT@font\MT@nl@ligatures
1875 \fi
1876 }

```

`\MT@noligatures@` This is also used by `\MT@set@tr@codes@`.

```

1877 \def\MT@noligatures@#1#2{%
1878   \MT@ifdefined@c@TF#2{%

```

Early MiK_T_EX versions (before 2.5.2579) didn't know `\tagcode`.

```

1879   \MT@ifdefined@c@TF\tagcode{%

```

No 'inputenc' key.

```

1880   \let\MT@warn@maybe@inputenc\@empty
1881   \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
1882   \MT@map@c@list@c#2{%
1883     \KV@sp@def\@tempa{##1}\MT@get@slot
1884     \ifnum\MT@char>\m@ne \tagcode#1\MT@char=\m@ne \fi}%
1885   \MT@vinfo{... Disabling ligatures for characters: #2}%
1886 }%
1887 \pdfnoligatures#1%
1888 \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
1889   know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
1890   the font instead}%
1891 }%
1892 }%

```

```

1893 \pdfnoligatures#1%
1894 \MT@vinfo{... Disabling ligatures}%
1895 }%
1896 }
1897 }\relax

```

14.2.7 Loading the configuration

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

```

1898 \def\MT@load@list#1{%
1899 \edef\@tempa{#1}%
1900 \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa @load}%
1901 \MT@ifstreq\@tempa\@tempb{%
1902 \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempa' cannot load itself}{}%
1903 }{%
1904 \ifx\@tempb\relax \else
1905 \MT@ifdefined@n@TF{MT@MT@feat @c@\@tempb}{%
1906 \MT@vinfo{... : First loading \@nameuse{MT@abbr@MT@feat} list `@\@tempb'%
1907 \begingroup
1908 \MT@load@list\@tempb
1909 \endgroup
1910 \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list
1911 \noexpand\MessageBreak`@\@tempb'}%
1912 \MT@let@cn\@tempc{MT@MT@feat @c@\@tempb}%
1913 \expandafter\MT@set@codes\@tempc,\relax,%
1914 }{%
1915 \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempb' undefined.\MessageBreak
1916 Cannot load it from list `@\@tempa'}{}%
1917 }%
1918 \fi
1919 }%
1920 }

```

`\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.

```

1921 \let\MT@file@list\@empty
1922 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

1923 \MT@in@clist{#1}\MT@file@list
1924 \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.

```

1925 \MT@begin@catcodes
1926 \let\MT@begin@catcodes\relax
1927 \let\MT@end@catcodes\relax
1928 \InputIfFileExists{mt-#1.cfg}{%
1929 \edef\MT@curr@file{mt-#1.cfg}%
1930 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1931 \MT@xadd\MT@file@list{#1,}%
1932 }{%
1933 \MT@get@basefamily#1\@empty\@empty\@empty\@nil
1934 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1935 \ifMT@inlist@
1936 \MT@xadd\MT@file@list{#1,}%
1937 \else
1938 \InputIfFileExists{mt-@\@tempa.cfg}{%
1939 \edef\MT@curr@file{mt-@\@tempa.cfg}%
1940 \MT@vinfo{... Loading configuration file \MT@curr@file}%

```

```

1941      \MT@xadd\MT@file@list{\@tempa,#1,}%
1942      }{%
1943      \MT@vinfo{... No configuration file mt-#1.cfg}%
1944      \MT@xadd\MT@file@list{#1,}%
1945      }%
1946      \fi
1947      }%
1948      \endgroup
1949      \fi
1950      }

```

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

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

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

```

1951 \def\MT@cfg@catcodes{%
1952   \makeatletter
1953   \catcode\^7%
1954   \catcode\ 9%
1955   \catcode\^^I9%
1956   \catcode\^^M9%
1957   \catcode\\\z@
1958   \catcode\{\@ne
1959   \catcode\}\@tw@
1960   \catcode\#6%
1961   \catcode\%14%
1962   \MT@map@tlist@n
1963   {\!\"$&\'\(\)\*+,\-\.\/\:\;\<=\>\?[\]\_-\`|\-}%
1964   \makeother
1965 }

```

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

```

1966 \def\MT@begin@catcodes{%
1967   \begingroup
1968   \MT@cfg@catcodes
1969 }

```

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

```

1970 \let\MT@end@catcodes\endgroup

```

`\MT@get@basefamily` The family name might have a suffix e. g., for expert set (x), old style numbers (j) swash capitals (w) etc. 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` ...).

We only work on the font name if it is longer than three characters.

```

1971 \def\MT@get@basefamily#1#2#3#4\@nil{%
1972   \ifx\@empty#4%
1973     \def\@tempa{#1#2#3}%
1974   \else
1975     \let\@tempa\@empty
1976     \edef\@tempb{#1#2#3#4}%
1977     \expandafter\MT@get@basefamily@\@tempb\@nil
1978   \fi
1979 }

```

Table 4: 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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

`\MT@get@basefamily@` This will only remove one suffix (the longest match), so that *combinations* of suffixes would have to be added manually (e. g., `\DeclareMicrotypeVariants*{aw}`). But otherwise, something like ‘padx’ would be truncated to ‘p’.

```

1980 \def\MT@get@basefamily@#1#2\@nil{%
1981   \edef\@tempa{\@tempa#1}%
1982   \ifx\#2\expandafter\@gobble\else\expandafter\@firstofone\fi
1983   {\MT@in@tlist{#2}\MT@variants
1984    \ifMT@in@tlist\else\MT@get@basefamily@#2\@nil\fi}%
1985 }
```

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

```

\MT@get@listname
\MT@get@listname@
1986 \def\MT@get@listname#1{%
1987   (debug)\MT@info{n1}{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
1988   \let\MT@listname\@undefined
1989   \def\@tempb{#1}%
1990   \MT@map@tlist@c\MT@try@order\MT@get@listname@
1991 }
1992 \def\MT@get@listname@#1{%
1993   \expandafter\MT@next@listname#1%
1994   \ifx\MT@listname\@undefined \else
1995     \expandafter\MT@tlist@break
1996   \fi
1997 }
```

`\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 4 in the documentation part any longer and can cast it off here.

```

1998 \def\MT@try@order{%
1999   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2000   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2001 }
```

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

```

2002 \def\MT@next@listname#1#2#3#4{%
2003   \edef\@tempa{\MT@encoding
2004     /\ifnum#1=\@ne \MT@family\fi
2005     /\ifnum#2=\@ne \MT@series\fi
2006     /\ifnum#3=\@ne \MT@shape\fi
2007     /\ifnum#4=\@ne *\fi
2008     \MT@context}%
2009   (debug)\MT@info{n1}{1}{trying \@tempa}%
2010   \MT@ifdefined@n@TF{MT@@tempb @\@tempa}{%
2011     \MT@next@listname@#4%
2012   }{%
```

Also try with an alias family.

```

2013   \ifnum#1=\@ne
2014     \ifx\MT@familyalias\@empty \else
2015       \edef\@tempa{\MT@encoding
```



```

2016          /\MT@familyalias
2017          /\ifnum#2=\@ne \MT@series\fi
2018          /\ifnum#3=\@ne \MT@shape\fi
2019          /\ifnum#4=\@ne *\fi
2020          \MT@context}%
2021 (debug)\MT@edinfo@n1{1}{(alias) \@tempa}%
2022          \MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%
2023          \MT@next@listname@#4%
2024          }%
2025          \fi
2026        \fi
2027      }%
2028    }

```

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

```

2029 \def\MT@next@listname@#1{%
2030   \ifnum#1=\@ne
2031     \MT@exp@cs\MT@in@list{MT@\@tempb @\@tempa @size}%
2032     \ifMT@in@list@
2033       \let\MT@listname\MT@size@name
2034     \fi
2035   \else
2036     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
2037   \fi
2038 }

```

\MT@if@list@exists

```

\MT@context 2039 \def\MT@if@list@exists{%
2040   \MT@let@cn\MT@context{MT@\MT@feat @context}%
2041   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
2042   \MT@get@listname{\MT@feat @c}%
2043   \MT@ifdefined@c@TF\MT@listname{%
2044     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
2045     \ifMT@nonselected
2046       \MT@vinfo{... Applying non-selected expansion (list '\MT@listname')}%
2047     \else
2048       \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list '\MT@listname'}%
2049     \fi
2050     \@firstoftwo
2051   }%

```

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

```
2052   \MT@let@nc{MT@\MT@feat @c@name}\@empty
```

Don't warn if selected=false.

```

2053   \ifMT@nonselected
2054     \MT@vinfo{... Applying non-selected expansion (no list)}%
2055   \else

```

Tracking doesn't require a list, either.

```

2056   \MT@ifstreq\MT@feat{tr}\relax{%
2057     \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
2058       for font\MessageBreak'\MT@@font'%
2059       \ifx\MT@context\@empty\else\space(context: '\MT@context')\fi.
2060     Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
2061   }%
2062   \fi
2063   \@secondoftwo
2064 }%
2065 }

```

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

```

\MT@context 2066 \def\MT@get@inh@list{%
2067   \let\MT@context\@empty
2068   \MT@get@listname{\MT@feat @inh}%
2069   \MT@ifdefined@c@TF\MT@listname{%
2070     \MT@edefn{MT@\MT@feat @inh@name}{\MT@listname}%
2071     (debug) \MT@info@n1{1}{... Using \nameuse{MT@abbr@\MT@feat} inheritance list
2072     (debug)             ~\MT@listname'}%
2073     \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%

```

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

```

2074   \ifx\@tempc\@empty \else
2075   (debug) \MT@info@n1{1}{parsing inheritance list ...}%

```

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

```

2076   \begingroup
2077   \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak~\MT@listname'}%
2078   \MT@set@inputenc{inh}%
2079   \expandafter\MT@inh@do\@tempc,\relax,%
2080   \MT@gl@et@nc{MT@\MT@feat @inh@\MT@listname}\@empty
2081   \endgroup
2082   \fi
2083   }{%
2084   \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
2085   }%
2086 }

```

14.2.8 Translating characters into slots

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

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

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

```

\MT@char@ 2087 \def\MT@get@slot{%
2088   \escapechar~\
2089   \let\MT@char@m@ne
2090   \MT@noresttrue

```

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

```

2091   \MT@toks=\expandafter{\@tempa}%

```

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

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

```

2092   \expandafter\MT@is@letter\@tempa\relax\relax
2093   \ifnum\MT@char@ < \z@

```

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

```

2094   \MT@exp@two@c\MT@is@active\string\@tempa\@nil

```

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

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

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

```
2095 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
2096 \MT@is@symbol
```

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

```
2097 {\expandafter\MT@is@composite\@tempa\relax\relax}%
2098 \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.

```
2099 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2100 \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2101 \fi
2102 \fi

2103 \let\MT@char\MT@char@
2104 \ifnum\MT@char@ < \z@
2105 \MT@warn@unknown
2106 \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.

```
2107 \ifMT@norest \else
2108 \MT@warn@rest
2109 \let\MT@char\m@ne
2110 \fi
2111 \fi
2112 \escapechar\m@ne
2113 }
```

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

```
2114 \newif\ifMT@norest
```

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

```
2115 \def\MT@is@letter#1#2\relax{%
2116 \ifcat a\noexpand#1\relax
2117 \edef\MT@char@{\number`#1}%
2118 \ifx\#2\%
2119 (debug)\MT@info{n}{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2120 \else
2121 \MT@norestfalse
2122 \fi
2123 \else
2124 \ifcat !\noexpand#1\relax
2125 \edef\MT@char@{\number`#1}%
2126 (debug)\MT@info{n}{3}{> `the\MT@toks' is a character (\MT@char@)}%
2127 \ifx\#2\%
2128 \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
2129 \else
2130 \MT@norestfalse
2131 \expandafter\MT@is@number#1#2\relax\relax
2132 \fi
2133 \fi
2134 \fi
```

```

2135 }
\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.
2136 \def\MT@is@number#1#2#3\relax{%
2137   \ifx\relax#3\relax \else
2138     \ifx\relax#2\relax \else
2139       \MT@noesttrue
2140       \if#1"\relax
2141         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2142   <debug>\MT@info@n1{3}{> ... a hexadecimal number: \MT@char@}%
2143     \else
2144       \if#1'\relax
2145         \def\MT@char@{\number#1#2#3}%
2146   <debug>\MT@info@n1{3}{> ... an octal number: \MT@char@}%
2147     \else
2148       \MT@ifint{#1#2#3}{%
2149         \def\MT@char@{\number#1#2#3}%
2150   <debug>\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
2151       } \MT@noestfalse
2152     \fi
2153   \fi
2154   \ifnum\MT@char@ > \@cclv
2155     \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2156     \let\MT@char@\m@ne
2157   \fi
2158   \fi
2159   \fi
2160 }

```

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

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

Unicode characters (`inputenc/utf8,utf8x`) are also supported.

```

2161 \def\MT@is@active#1#2\@nil{%
2162   \ifnum\catcode#1 = \active
2163     \begingroup
2164     \set@display@protect
2165     \let\IeC\@firstofone
2166     \let\@inpenc@undefined@\MT@undefined@char

```

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

```

2167   \def\UTFviii@defined##1{\ifx ##1\relax
2168     \MT@undefined@char{utf8}\else\expandafter ##1\fi}%

```

For `ucs (utf8x)`. Let's call it experimental ...

```

2169   \MT@ifdefined@c@T\PrerenderUnicode
2170   {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
2171   \edef\x{\endgroup
2172     \def\noexpand\@tempa{\@tempa}%

```

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

```
2173 \MT@toks={\the\MT@toks\space(=\@tempa)}%
2174 }%
2175 \x
2176 \fi
2177 }
```

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

```
2178 \def\MT@undefined@char#1{undefined in input encoding ``#1''}
```

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

```
2179 \def\MT@is@symbol{%
2180 \expandafter\def\expandafter\MT@char\expandafter
2181 {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2182 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2183 \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2184 \ifnum\MT@char@ < \z@
```

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

```
2185 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2186 \fi
2187 }
```

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

```
\MT@charstring 2188 \begingroup
2189 \catcode`\=/\z@
2190 /MT@map@tlist@n{/CHAR}/@makeoother
2191 /lowercase{%
2192 /def/x{/endgroup
2193 /def/MT@charstring{CHAR}%
2194 /def/MT@is@char##1CHAR"##2##3##4/relax{%
2195 /ifx/relax##1/relax
2196 /if##3/relax
2197 /edef/MT@char@{/number"##2}%
2198 /MT@ifstreq/MT@charstring{##3##4}/relax/MT@norestfalse
2199 /else
2200 /edef/MT@char@{/number"##2##3}%
2201 /MT@ifstreq/MT@charstring{##4}/relax/MT@norestfalse
2202 /fi
2203 <debug> /MT@info@n1{3}{> ~/the/MT@toks' is a \char (/MT@char@)}%
2204 /fi
2205 }%
2206 }%
2207 }
2208 /x
```

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

```
2209 \def\MT@is@composite#1#2\relax{%
2210 \ifx\#2\\\else
```

Again, we construct a control sequence, this time of the form: `\<encoding>\<accent>-<character>`, e. g., `\T1"-a`, which we then expand once to see if it is a letter (if it has been defined by `\DeclareTextComposite`). This should be robust, finally, especially, since we also `\detokenize` the input instead of only `\stringifying`

it. Thus, we will die gracefully even on wrong Unicode input without utf8.

```

2211 \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2212 \string\csname\MT@encoding\endcsname
2213 \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
2214 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2215 \fi
2216 }

```

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

```

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

```

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

Some warning messages, for performance reasons separated here.

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

```

\MT@set@listname 2217 \def\MT@set@listname{%
2218 \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
2219 \~\@nameuse{MT@\MT@feat @c@name}}}%
2220 }

```

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

```

2221 \def\MT@warn@ascii{%
2222 \MT@warning@nl{Character '\the\MT@toks' (= \MT@char@)
2223 is outside of ASCII range.\MessageBreak
2224 You must load the 'inputenc' package before using\MessageBreak
2225 8-bit characters in \MT@curr@list@name}%
2226 }

```

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

```

2227 \def\MT@warn@number@too@large#1{%
2228 \MT@warning@nl{%
2229 Number #1 in encoding '\MT@encoding' too large!\MessageBreak
2230 Ignoring it in \MT@curr@list@name}%
2231 }

```

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

```

2232 \def\MT@warn@rest{%
2233 \MT@warning@nl{%
2234 Unknown slot number of character\MessageBreak'\the\MT@toks'%
2235 \MT@warn@maybe@inputenc\MessageBreak
2236 in font encoding '\MT@encoding'.\MessageBreak
2237 Make sure it's a single character\MessageBreak
2238 (or a number) in \MT@curr@list@name}%
2239 }

```

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

```
2240 \def\MT@warn@unknown{%
2241   \MT@warning@nl{%
2242     Unknown slot number of character\MessageBreak`the\MT@toks'%
2243     \MT@warn@maybe@inputenc\MessageBreak
2244     in font encoding `\'MT@encoding' in \MT@curr@list@name}%
2245 }
```

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

```
2246 \def\MT@warn@maybe@inputenc{%
2247   \MT@ifdefined@n@T
2248   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
2249   { (input encoding `\'@nameuse
2250    {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2251 }
```

14.2.9 Hook into L^AT_EX's font selection

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

In contrast to the `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 my reference:

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

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

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

```
\MT@font 2252 \let\MT@font@list\@empty
2253 \let\MT@font\@empty
```

All this is done at the beginning of the document. It doesn't work for plain, of course, which doesn't have `\pickup@font`.

```
2254 </package>
2255 <plain>\MT@requires@latex2{
2256 \MT@addto@setup{%
```

`\MT@orig@pickupfont` microtype also seems to work with CJK.

```
2257 \ifpackageloaded{CJK}{
2258 \ifpackageafter{CJK}{2006/10/17}{% 4.7.0
2259 {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}}
2260 {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}}
2261 \g@addto@macro\MT@orig@pickupfont
2262 {\expandafter\ifx\font@name\relax\define@newfont\fi}}
```

CJKutf8 redefines `\pickup@font` once more (recent versions, in PDF mode, as determined by `ifpdf`, which CJKutf8 loads).

```
2263 \ifpackageloaded{CJKutf8}
2264 {\ifpackageafter{CJKutf8}{2008/05/22}{% 4.8.0
2265 {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}
2266 {\@firstoftwo}}
2267 {\@firstoftwo}}
2268 {\g@addto@macro\MT@orig@pickupfont{
2269 {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2270 \define@newfont\else\xdef\font@name{
2271 \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}
2272 {\g@addto@macro\MT@orig@pickupfont{
2273 {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2274 \define@newfont\def\CJK@temp{v}%
2275 \ifx\CJK@temp\CJK@plane
2276 \expandafter\ifx\csname CJK@cmap@\f@family\CJK@plane\endcsname\relax
2277 \else\csname CJK@cmap@\f@family\CJK@plane\endcsname\fi
2278 \else \CJK@addcmap\CJK@plane \fi
2279 \else\xdef\font@name{
2280 \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}
2281 }}
2282 {\def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
2283 }
```

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

```
2284 \ifx\pickup@font\MT@orig@pickupfont \else
2285 \MT@warning@nl{%
2286 Command \string\pickup@font\space is not defined as expected.%
2287 \MessageBreak Patching it anyway. Some things may break%
2288 <*package>
2289 .\MessageBreak Double-check whether micro-typography is indeed%
2290 \MessageBreak applied to the document.%
2291 \MessageBreak (Hint: Turn on `verbose' mode)%
2292 </package>
2293 }
2294 \fi
```

`\pickup@font` Then we append our stuff. Everything is done inside a group.

```
2295 \g@addto@macro\pickup@font{\begingroup}
```

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

extremely noisy.

```

2296 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
2297 \g@addto@macro\pickup@font{%
2298   \escapechar\m@ne
2299 }*package
2300 <debug> \global\MT@inannottrue
2301 <debug> \MT@glet\MT@pdf@annot\@empty
2302 <debug> \MT@addto@annot{(line \number\inputlineno)}%
```

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

```

2303 \MT@let@cn\MT@font\MT@subst@expandafter\string\font@name}%
2304 \ifx\MT@font\relax
2305   \let\MT@font\font@name
2306 \else
2307   \ifx\MT@font\font@name \else
2308 <debug> \MT@addto@annot{= substituted with \MT@font}%
2309   \MT@register@subst@font
2310   \fi
2311 \fi
2312 \MT@setupfont
2313 </package>
2314 <letterspace> \MT@tracking
2315 \endgroup
2316 }
2317 <*package>
```

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

```
2318 \let\MT@pickupfont\pickup@font
```

`\do@subst@correction` 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. We have to remember the substitute for the rest of the document, not just for the first time it is called, since we need it every time a font is letterspaced.

```

2319 \g@addto@macro\do@subst@correction
2320 { \edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2321   \MT@glet@nc\MT@subst@expandafter\string\font@name\MT@font}
```

`\add@accent` Inside `\add@accent`, we have to disable `microtype`'s setup, since the grouping in
`\MT@orig@add@accent` 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.

```

2322 \let\MT@orig@add@accent\add@accent
2323 \def\add@accent#1#2{%
2324   \let\pickup@font\MT@orig@pickupfont
2325   \MT@orig@add@accent{#1}{#2}%
2326   \let\pickup@font\MT@pickupfont
2327 }
2328 </package>
2329 }
2330 <plain> \relax
2331 <*package>
```

Consequently, we are (if all goes well) the last ones to change these commands, therefore there is no need to check whether our definition has survived.

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

```
2332 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

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

```
2333 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
```

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

```
2334 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}
```

14.2.10 Context-sensitive setup

Here are the variants for context-sensitive setup.

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

```
2335 \let\MT@active@features\empty
```

`\MT@check@font@cx` Every feature has its own list of fonts that have already been dealt with. If the font needn't be set up for a feature, we temporarily disable the corresponding setup command. This should be more efficient than book-keeping the fonts in lists associated with the combination of contexts, as we've done it before.

```
2336 \def\MT@check@font@cx{%
2337   \MT@if@true
2338   \MT@map@clist@c\MT@active@features{%
2339     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
2340     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
2341     \ifMT@inlist@
2342       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
2343     \else
2344       \MT@if@false
2345     \fi
2346   }%
2347   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2348 }
```

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

```
2349 \def\MT@register@subst@font@cx{%
2350   \MT@map@clist@c\MT@active@features{%
2351     \MT@exp@cs\MT@xadd
2352     {MT@##1\csname MT@##1@context\endcsname font@list}%
2353     {\font@name,}%
2354   }%
2355 }
```

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

```
2356 \def\MT@register@font@cx{%
2357   \MT@map@clist@c\MT@active@features{%
2358     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
2359     \MT@exp@cs\MT@xadd
2360     {MT@##1\csname MT@##1@context\endcsname font@list}%
2361     {\MT@font,}%
2362     \def\@tempa{##1}%
2363     \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
2364   \fi
2365   }%
2366 }
```

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

```
2367 \def\MT@maybe@rem@from@list#1{%
```

```

2368 \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
2369 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
2370 \MT@font \csname MT@\@tempa @#1font@list\endcsname
2371 }%
2372 }

```

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

Inside the preamble, it shouldn't actually do anything but remember it for later.

```

2373 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
2374 \MT@addto@setup{%
2375 \DeclareRobustCommand\microtypecontext[1]{%
2376 \MT@setup@contexts
2377 \let\MT@reset@context\relax
2378 \setkeys{MTC}{#1}%
2379 \selectfont
2380 \MT@reset@context
2381 }%
2382 }

```

`\textmicrotypecontext` This is just a wrapper around `\microtypecontext`.

```

2383 \DeclareRobustCommand\textmicrotypecontext[2]{\microtypecontext{#1}#2}

```

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

`\MT@reset@context@`

```

2384 \def\MT@reset@context@{%
2385 \MT@vinfo{<<< Resetting contexts\on@line
2386 <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
2387 <debug> \MT@tr@context/\MT@kn@context/\MT@sp@context
2388 }%
2389 \selectfont
2390 }

```

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

```

2391 \def\MT@setup@contexts{%
2392 \MT@map@clist@c\MT@active@features
2393 {\MT@glet@c{MT@##1@font@list}\MT@font@list}%
2394 \MT@glet\MT@check@font\MT@check@font@cx
2395 \MT@glet\MT@register@font\MT@register@font@cx
2396 \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
2397 \MT@glet\MT@setup@contexts\relax
2398 }

```

Define context keys.

```

2399 \MT@map@clist@c\MT@features@long{%
2400 \define@key{MTC}{#1}[]{}%
2401 \edef\@tempb{\@nameuse{MT@rbba#1}}%
2402 \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2403 \ifMT@inlist@

```

Using an empty context is only asking for trouble, therefore we choose the '0' instead (hoping for the L^AT_EX users' natural awe of this character).

```

2404 \MT@ifempty{#1}{\def\MT@val{0}}{\def\MT@val{#1}}%
2405 \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
2406 <debug>\MT@edinfo{1}{>>> no change of #1 context: `'\MT@val'}%
2407 \else
2408 \MT@vinfo{>>> Changing #1 context to `'\MT@val'\MessageBreak\on@line
2409 <debug> \space(previous: `'\@nameuse{MT@\@tempb @context}')}%
2410 }%
2411 \def\MT@reset@context{\aftergroup\MT@reset@context@}%

```

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

```
2412 \MT@glet@nn{MT@reset@{\@tempb @codes}{MT@reset@{\@tempb @codes@}%}
```

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

```
2413 \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
2414 \MT@val \csname MT@\@tempb @doc@contexts\endcsname
2415 \ifMT@inlist@ \else
2416 \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{\MT@val}}%
2417 (debug) \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
2418 \fi
2419 \MT@edef@n{MT@\@tempb @context}{\MT@val}%
2420 \fi
2421 \fi
2422 }%
2423 }
```

\MT@pr@context Initialise the contexts.

\MT@ex@context 2424 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%}

\MT@tr@context 2425 \MT@def@n{MT@#1@context}{@}%

\MT@sp@context 2426 \MT@def@n{MT@#1@doc@contexts}{@}%

\MT@kn@context 2427 }

\MT@kn@context 2428 \let\MT@extra@context\@empty

\MT@pr@doc@contexts

\MT@ex@doc@contexts

\MT@tr@doc@contexts

\MT@sp@doc@contexts

\MT@kn@doc@contexts

\DeclareMicrotypeSet

\MT@extra@context

\DeclareMicrotypeSet*

14.3 Configuration

14.3.1 Font sets

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

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

```
2429 \def\DeclareMicrotypeSet{%
2430 \ifstar
2431 \MT@DeclareSetAndUseIt
2432 \MT@DeclareSet
2433 }

\MT@DeclareSet
2434 \newcommand\MT@DeclareSet[3][]{%
2435 \KV@sp@def\@tempa{#1}%
2436 \MT@ifempty\@tempa{%
2437 \MT@map@clist@c\MT@features{\MT@declare@sets{##1}{#2}{#3}}}%
2438 }{%
2439 \MT@map@clist@c\@tempa{%
2440 \KV@sp@def\@tempa{##1}%
2441 \MT@ifempty\@tempa\relax{%
2442 \MT@is@feature{set declaration `#2'}{%
2443 \MT@exp@one@n\MT@declare@sets
2444 {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
2445 }%
2446 }%
2447 }}%
2448 }%
2449 }
```

\MT@DeclareSetAndUseIt

```
2450 \newcommand\MT@DeclareSetAndUseIt[3][]{%}
```

```

2451 \MT@DeclareSet[#1]{#2}{#3}%
2452 \UseMicrotypeSet[#1]{#2}%
2453 }

```

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

```

2454 \let\MT@curr@set@name\empty

```

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

```

2455 \def\MT@declare@sets#1#2#3{%
2456 \KV@sp@def\MT@curr@set@name{#2}%
2457 \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
2458 \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2459 \MT@gl@et@nc{MT@#1@list@size@\MT@curr@set@name}\empty
2460 }%
2461 \MT@gl@et@nc{MT@#1@set@@\MT@curr@set@name}\empty
2462 <debug>\MT@info{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2463 \setkeys{MT@#1@set}{#3}%
2464 }

```

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

```

2465 \def\MT@define@set@key@#1#2{%
2466 \define@key{MT@#2@set}{#1}[]{}%
2467 \MT@gl@et@nc{MT@#2@list@#1@\MT@curr@set@name}\empty
2468 \MT@map@clist@n{##1}{%
2469 \KV@sp@def\MT@val{###1}%
2470 \MT@get@highlevel{#1}%

```

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

```

2471 \MT@exp@two@n@g@addto@macro
2472 {\csname MT@#2@list@#1@\MT@curr@set@name\expandafter\endcsname}%
2473 {\MT@val,}%
2474 }%

```

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

```

2475 \expandafter\g@addto@macro\expandafter\MT@font@sets
2476 \csname MT@#2@list@#1@\MT@curr@set@name\endcsname
2477 <debug>\MT@info@n1{1}{-- #1: \@nameuse{MT@#2@list@#1@\MT@curr@set@name}}%
2478 }%
2479 }

```

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

```

2480 \def\MT@get@highlevel#1{%
2481 \expandafter\MT@test@ast\MT@val*\@nil\relax{}%

```

And ‘family = *’ will become \familydefault.

```

2482 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
2483 \edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}%

```

In contrast to earlier version, these values will not be expanded immediately but at the end of the preamble.

```

2484 }%
2485 }

```

\MT@test@ast It the last character is an asterisk, execute the second argument, otherwise the first one.

```

2486 \def\MT@test@ast#1*#2\@nil{%
2487 \def\@tempa{#1}%
2488 \MT@ifempty{#2}%
2489 }

```

`\MT@font@sets` Fully expand the font specification and fix catcodes for all font sets.

```
\MT@fix@font@set 2490 \let\MT@font@sets\empty
2491 \def\MT@fix@font@set#1{%
2492   \xdef#1{#1}%
2493   \global\@onelevel@sanitize#1%
2494 }
```

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

```
2495 \def\MT@define@set@key@size#1{%
2496   \define@key{MT@#1@set}{size}[]{%
2497     \MT@map@clist@n{##1}%
2498     \KV@esp@def\MT@val{###1}%
2499     \expandafter\MT@get@range\MT@val--\@nil
2500     \ifx\MT@val\relax \else
2501       \MT@exp@cs\MT@xadd
2502       {MT@#1list@size@MT@curr@set@name}%
2503       {{{\MT@lower}{\MT@upper}\relax}}%
2504     \fi
2505   }%
2506   <debug>\MT@info@n1{1}{-- size: \@nameuse{MT@#1list@size@MT@curr@set@name}}%
2507   }%
2508 }
```

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

`\MT@get@range` Ranges will be stored as triplets of `{\lower bound}{\upper bound}{\list name}`.

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

```
\MT@lower 2509 \def\MT@get@range#1-#2-#3\@nil{%
2510   \MT@ifempty{#1}%
2511   \MT@ifempty{#2}%
2512   \let\MT@val\relax
2513   }{%
2514     \def\MT@lower{0}%
2515     \def\MT@val{#2}%
2516     \MT@get@size
2517     \edef\MT@upper{\MT@val}%
2518   }%
2519   }{%
2520     \def\MT@val{#1}%
2521     \MT@get@size
2522     \ifx\MT@val\relax \else
2523       \edef\MT@lower{\MT@val}%
2524       \MT@ifempty{#2}%
2525       \MT@ifempty{#3}%
2526       {\def\MT@upper{-1}}%
```

2048pt is \TeX 's maximum font size.

```
2527   {\def\MT@upper{2048}}%
2528   }{%
2529     \def\MT@val{#2}%
2530     \MT@get@size
2531     \ifx\MT@val\relax \else
2532       \MT@ifdim\MT@lower>\MT@val{%
2533         \MT@error{%
2534           Invalid size range (\MT@lower\space > \MT@val) in font set
2535           '\MT@curr@set@name'.\MessageBreak Swapping sizes}}%
2536       \edef\MT@upper{\MT@lower}%
```

```

2537         \edef\MT@lower{\MT@val}%
2538     }{%
2539         \edef\MT@upper{\MT@val}%
2540     }%
2541     \MT@ifdim\MT@lower=\MT@upper
2542     {\def\MT@upper{-1}}%
2543     \relax
2544 \fi
2545 }%
2546 \fi
2547 }%
2548 }

```

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

```

2549 \def\MT@get@size{%

```

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

```

2550     \if*\MT@val\relax
2551         \def\@tempa{\normalsize}%
2552     \else
2553         \MT@let@cn\@tempa{\MT@val}%
2554     \fi
2555     \ifx\@tempa\relax \else

```

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

```

2556     \begingroup
2557     \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
2558     \@tempa\@nil
2559 \fi

```

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

```

2560 \MT@ifdimen\MT@val{%
2561     \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2562     \edef\MT@val{\strip@pt\@tempdima}%
2563 }{%
2564     \MT@warning{Could not parse font size `~\MT@val'\MessageBreak
2565         in font set `~\MT@curr@set@name'}%
2566     \let\MT@val\relax
2567 }%
2568 }

```

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

```

2569 \def\MT@define@set@key@font#1{%
2570     \define@key{MT@#1@set}{font}[]{}%
2571     \MT@gl@et@nc{MT@#1list@font@\MT@curr@set@name}\@empty
2572     \MT@map@clist@n{##1}{%
2573         \KV@sp@def\MT@val{###1}%
2574         \MT@ifstreq\MT@val*\def\MT@val{*/*/*/*}\relax
2575         \expandafter\MT@get@font\MT@val////\@nil
2576         \MT@exp@two@n@g@addto@macro
2577             {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
2578             {\MT@val,}%
2579     }%
2580     \expandafter\g@addto@macro\expandafter\MT@font@sets
2581     \csname MT@#1list@font@\MT@curr@set@name\endcsname
2582     <debug>\MT@edinfo@n{1}{-- font: \nameuse{MT@#1list@font@\MT@curr@set@name}}%

```

```
2583 }%
2584 }
```

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

```
2585 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2586 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
2587 \ifx\MT@val\relax\def\MT@val{0}\fi
2588 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2589 \let\MT@val\@tempb
2590 }
```

`\MT@get@font@` Helper macro, also used by `\MT@get@font@and@size`.

```
2591 \def\MT@get@font@#1#2#3#4#5#6{%
2592 \let\@tempb\@empty
2593 \def\MT@temp{#1/#2/#3/#4/#5}%
2594 \MT@get@axis{encoding}{#1}%
2595 \MT@get@axis{family}{#2}%
2596 \MT@get@axis{series}{#3}%
2597 \MT@get@axis{shape}{#4}%
2598 \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
2599 \MT@ifempty{#5}{%
2600 \MT@warn@axis@empty{size}{\string\normalsize}%
2601 \def\MT@val{*}%
2602 }{%
2603 \def\MT@val{#5}%
2604 }%
2605 \MT@get@size
2606 }
```

`\MT@get@axis`

```
2607 \def\MT@get@axis#1#2{%
2608 \def\MT@val{#2}%
2609 \MT@get@highlevel{#1}%
2610 \MT@ifempty\MT@val{%
2611 \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2612 \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2613 }\relax
2614 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2615 }
```

`\MT@warn@axis@empty`

```
2616 \def\MT@warn@axis@empty#1#2{%
2617 \MT@warning{#1 axis is empty in font specification\MessageBreak
2618 ~\MT@temp'. Using ~#2' instead}%
2619 }
```

We have finally assembled all pieces to define `\DeclareMicrotypeSet`'s keys. It is also used for `\DisableLigatures`.

```
2620 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
2621 \MT@define@set@key@{encoding}{#1}%
2622 \MT@define@set@key@{family}{#1}%
2623 \MT@define@set@key@{series}{#1}%
2624 \MT@define@set@key@{shape}{#1}%
2625 \MT@define@set@key@size{#1}%
2626 \MT@define@set@key@font{#1}%
2627 }
```

`\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.

```
2628 \renewcommand*\UseMicrotypeSet[2][{}]{%
2629 \KV@sp@def\@tempa{#1}%

```



```

2630 \MT@ifempty\@tempa{%
2631 \MT@map@clist@c\MT@features{\MT@use@set{##1}{#2}}}%
2632 }{%
2633 \MT@map@clist@c\@tempa{%
2634 \KV@sp@def\@tempa{##1}%
2635 \MT@ifempty\@tempa\relax{%
2636 \MT@is@feature{activation of set `#2'}{%
2637 \MT@exp@one@n\MT@use@set
2638 {\csname MT@rbba@\@tempa\endcsname}{#2}%
2639 }%
2640 }%
2641 }}%
2642 }%
2643 }

```

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

```

\MT@ex@setname 2644 \def\MT@use@set#1#2{%
\MT@tr@setname 2645 \KV@sp@def\@tempa{#2}%
2646 \MT@ifdefined@n@TF{MT@#1@set@\@tempa}{%
\MT@sp@setname 2647 \MT@xdef@n{MT@#1@setname}{\@tempa}%
\MT@kn@setname 2648 }{%
\MT@use@set 2649 \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2650 \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2651 }%
2652 \MT@error{%
2653 The \@nameuse{MT@abbr@#1} set `@\tempa' is undeclared.\MessageBreak
2654 Using set `@\nameuse{MT@#1@setname}' instead}{}%
2655 }%
2656 }

```

\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.

```

2657 \renewcommand*{\DeclareMicrotypeSetDefault}[2][]{%
2658 \KV@sp@def\@tempa{#1}%
2659 \MT@ifempty\@tempa{%
2660 \MT@map@clist@c\MT@features{\MT@set@default@set{##1}{#2}}}%
2661 }{%
2662 \MT@map@clist@c\@tempa{%
2663 \KV@sp@def\@tempa{##1}%
2664 \MT@ifempty\@tempa\relax{%
2665 \MT@is@feature{declaration of default set `#2'}{%
2666 \MT@exp@one@n\MT@set@default@set
2667 {\csname MT@rbba@\@tempa\endcsname}{#2}%
2668 }%
2669 }%
2670 }}%
2671 }%
2672 }

```

\MT@default@pr@set

```

\MT@default@ex@set 2673 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 2674 \KV@sp@def\@tempa{#2}%
2675 \MT@ifdefined@n@TF{MT@#1@set@\@tempa}{%
\MT@default@sp@set 2676 (debug)\MT@info{1}{declaring default \@nameuse{MT@abbr@#1} set `@\tempa'}%
\MT@default@kn@set 2677 \MT@xdef@n{MT@default@#1@set}{\@tempa}%
\MT@set@default@set 2678 }{%
2679 \MT@error{%
2680 The \@nameuse{MT@abbr@#1} set `@\tempa' is not declared.\MessageBreak
2681 Cannot make it the default set. Using set\MessageBreak `all' instead}{}%
2682 \MT@xdef@n{MT@default@#1@set}{all}%
2683 }%
2684 }

```

14.3.2 Variants and aliases

`\DeclareMicrotypeVariants` Specify suffixes for variants (see `fontname/variants.map`). The starred version `\MT@variants` appends to the list.

```
2685 \let\MT@variants\@empty
2686 \def\DeclareMicrotypeVariants{%
2687   \ifstar
2688     \MT@DeclareVariants
2689     {\let\MT@variants\@empty\MT@DeclareVariants}%
2690 }
```

`\MT@DeclareVariants`

```
2691 \def\MT@DeclareVariants#1{%
2692   \MT@map@clist@n{#1}%
2693   \KV@sp@def\@tempa{##1}%
2694   \@onelevel@sanitize\@tempa
2695   \xdef\MT@variants{\MT@variants{\@tempa}}%
2696   }%
2697 }
```

`\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.

```
2698 \renewcommand*\DeclareMicrotypeAlias[2]{%
2699   \KV@sp@def\@tempa{#1}%
2700   \KV@sp@def\@tempb{#2}%
2701   \@onelevel@sanitize\@tempb
2702   \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2703     \MT@warning{Alias font family '\@tempb' will override
2704       alias \@nameuse{MT@\@tempa @alias}'\MessageBreak
2705       for font family '\@tempa'}}%
2706   \MT@xdef@n{MT@\@tempa @alias}{\@tempb}%
```

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

```
2707 \MT@ifdefined@c@T{MT@family{%
2708   <debug>\MT@dinfo{1}{Activating alias font '\@tempb' for '\MT@family'}%
2709   \MT@glet\MT@familyalias\@tempb
2710   }%
2711 }
```

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

```
2712 \def\LoadMicrotypeFile#1{%
2713   \KV@sp@def\@tempa{#1}%
2714   \@onelevel@sanitize\@tempa
2715   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2716   \ifMT@inlist@
2717     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2718   \else
2719     \MT@xadd\MT@file@list{\@tempa,}%
2720     \MT@begin@catcodes
2721     \InputIfFileExists{mt-\@tempa.cfg}%
2722     \edef\MT@curr@file{mt-\@tempa.cfg}%
2723     \MT@vinfo{... Loading configuration file \MT@curr@file}%
2724     }{%
2725     \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2726       does not exist}%
2727     }%
2728     \MT@end@catcodes
2729   \fi
2730 }
```

14.3.3 Disabling ligatures

`\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'.

`\MT@DisableLigatures`

`\MT@nl@setname` The optional argument may be used to disable selected ligatures only.

```

\MT@nl@ligatures 2731 \MT@requires@pdftex5{
2732 \def\DisableLigatures{%
2733 \MT@begin@catcodes
2734 \MT@DisableLigatures
2735 }
2736 \newcommand*\MT@DisableLigatures[2][]{%
2737 \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
2738 \xdef\MT@active@features{\MT@active@features,nl}%
2739 \global\MT@no_ligaturestrue
2740 \MT@declare@sets{nl}{no ligatures}{#2}%
2741 \gdef\MT@nl@setname{no ligatures}%
2742 \MT@end@catcodes
2743 }
2744 }{
    If pdfTeX is too old, we throw an error.
2745 \renewcommand*\DisableLigatures[2][]{%
2746 \MT@error{Disabling ligatures of a font is only possible\MessageBreak
2747 with pdftex version 1.30 or newer.\MessageBreak
2748 Ignoring \string\DisableLigatures}{Upgrade pdftex.}%
2749 }
2750 }

```

14.3.4 Interaction with babel

`\DeclareMicrotypeBabelHook` Declare the context that should be loaded when a babel language is selected. The command will not check whether a previous declaration will be overwritten.

```

2751 \def\DeclareMicrotypeBabelHook#1#2{%
2752 \MT@map@clist@n{#1}{%
2753 \KV@esp@def\@tempa{##1}%
2754 \MT@gdef@n{\MT@babel@\@tempa}{#2}%
2755 }%
2756 }

```

14.3.5 Fine tuning

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

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

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

```

2757 \def\SetProtrusion{%
2758 \MT@begin@catcodes
2759 \MT@SetProtrusion
2760 }

```

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

```

\MT@pr@c@name 2761 \newcommand*\MT@SetProtrusion[3][]{%
\MT@extra@context 2762 \let\MT@extra@context\@empty
\MT@permutelist

```

Parse the optional first argument. We first have to know the name before we can deal with the extra options.

```
2763 \MT@set@named@keys{MT@pr@c}{#1}%
2764 (debug)\MT@info{1}{creating protrusion list `\'MT@pr@c@name'}%
2765 \def\MT@permutelist{pr@c}%
2766 \setkeys{MT@cfg}{#2}%
```

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

```
2767 \MT@permute
```

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

```
2768 \MT@gdef@n{MT@pr@c@\'MT@pr@c@name}{#3}%
2769 \MT@end@catcodes
2770 }
```

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

```
2771 \def\SetExpansion{%
2772 \MT@begin@catcodes
2773 \MT@SetExpansion
2774 }
```

\MT@SetExpansion

```
\MT@ex@c@name 2775 \newcommand*\'MT@SetExpansion[3] [] {%
\MT@extra@context 2776 \let\MT@extra@context\@empty
\MT@permutelist 2777 \MT@set@named@keys{MT@ex@c}{#1}%
2778 \MT@ifdefined@n@T{MT@ex@c@\'MT@ex@c@name @factor}{%
2779 \ifnum\csize MT@ex@c@\'MT@ex@c@name @factor\endcsize > \@m
2780 \MT@warning@n1{Expansion factor \number\@nameuse{MT@ex@c@\'MT@ex@c@name @factor}
2781 too large in list\MessageBreak `\'MT@ex@c@name'. Setting it to the
2782 maximum of 1000}%
2783 \MT@glet@nc{MT@ex@c@\'MT@ex@c@name @factor}\@m
2784 \fi
2785 }%
2786 (debug)\MT@info{1}{creating expansion list `\'MT@ex@c@name'}%
2787 \def\MT@permutelist{ex@c}%
2788 \setkeys{MT@cfg}{#2}%
2789 \MT@permute
2790 \MT@gdef@n{MT@ex@c@\'MT@ex@c@name}{#3}%
2791 \MT@end@catcodes
2792 }
```

\SetTracking

```
2793 \def\SetTracking{%
2794 \MT@begin@catcodes
2795 \MT@SetTracking
2796 }
```

\MT@SetTracking Third argument may be empty.

```
2797 \newcommand*\'MT@SetTracking[3] [] {%
2798 \let\MT@extra@context\@empty
2799 \MT@set@named@keys{MT@tr@c}{#1}%
2800 (debug)\MT@info{1}{creating tracking list `\'MT@tr@c@name'}%
2801 \def\MT@permutelist{tr@c}%
2802 \setkeys{MT@cfg}{#2}%
2803 \MT@permute
2804 \KV@sp@def\@tempa{#3}%
2805 \MT@ifempty\@tempa\relax{%
2806 \MT@ifint\@tempa
```

```

2807      {\MT@xdef@n{MT@tr@cc@MT@tr@cc@name}{\@tempa}}%
2808      {\MT@warning{Value `~\@tempa' is not a number in\MessageBreak
2809                tracking set `~\MT@curr@set@name'}}}%
2810      \MT@end@catcodes
2811  }

\SetExtraSpacing
2812 \def\SetExtraSpacing{%
2813   \MT@begin@catcodes
2814   \MT@SetExtraSpacing
2815  }

\MT@SetExtraSpacing
\MT@sp@c@name 2816 \newcommand*\MT@SetExtraSpacing[3] [] {%
\MT@extra@context 2817   \let\MT@extra@context\@empty
\MT@set@named@keys{MT@sp@c}{#1}%
\MT@permute!ist 2818   \MT@set@named@keys{MT@sp@c}{#1}%
2819   (debug)\MT@dinfor{1}{creating spacing list `~\MT@sp@c@name'}%
2820   \def\MT@permute!ist{sp@c}%
2821   \setkeys{MT@cfg}{#2}%
2822   \MT@permute
2823   \MT@gdef@n{MT@sp@c@MT@sp@c@name}{#3}%
2824   \MT@end@catcodes
2825  }

\SetExtraKerning
2826 \def\SetExtraKerning{%
2827   \MT@begin@catcodes
2828   \MT@SetExtraKerning
2829  }

\MT@SetExtraKerning
\MT@kn@c@name 2830 \newcommand*\MT@SetExtraKerning[3] [] {%
\MT@extra@context 2831   \let\MT@extra@context\@empty
2832   \MT@set@named@keys{MT@kn@c}{#1}%
\MT@permute!ist 2833   (debug)\MT@dinfor{1}{creating kerning list `~\MT@kn@c@name'}%
2834   \def\MT@permute!ist{kn@c}%
2835   \setkeys{MT@cfg}{#2}%
2836   \MT@permute
2837   \MT@gdef@n{MT@kn@c@MT@kn@c@name}{#3}%
2838   \MT@end@catcodes
2839  }

\MT@set@named@keys We first set the name (if specified), then remove it from the list, and set the
\MT@options remaining keys.
2840 \def\MT@set@named@keys#1#2{%
2841   \def\x##1name=##2,##3\@nil{%
2842     \setkeys{#1}{name=##2}%
2843     \gdef\MT@options{##1##3}%
2844     \MT@rem@from@clist{name=}\MT@options
2845   }%
2846   \x#2,name=,\@nil
2847   \@expandtwoargs\setkeys{#1}\MT@options
2848  }

\MT@define@code@key Define the keys for the configuration lists (which are setting the codes, in pdfTeX
speak).
2849 \def\MT@define@code@key#1#2{%
2850   \define@key{MT@#2}{#1} [] {%
2851     \@tempcnta=\@ne
2852     \MT@map@clist@n{##1}%
2853     \KV@sp@def\MT@val{###1}%

```

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

```

2854 \MT@get@highlevel{#1}%
2855 \MT@edef\MT@temp#1\the\@tempcnta\MT@val}%
2856 \advance\@tempcnta \@ne
2857 }%
2858 }%
2859 }

```

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

```

2860 \def\MT@define@code@key@size#1{%
2861 \define@key{MT@#1}{size}[]{%
2862 \MT@map@clist@n{##1}%
2863 \KV@sp@def\MT@val{###1}%
2864 \expandafter\MT@get@range\MT@val--\@nil
2865 \ifx\MT@val\relax \else
2866 \MT@exp@cs\MT@xadd\MT@tempsize{%
2867 {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2868 \fi
2869 }%
2870 }%
2871 }

```

\MT@define@code@key@font

```

2872 \def\MT@define@code@key@font#1{%
2873 \define@key{MT@#1}{font}[]{%
2874 \MT@map@clist@n{##1}%
2875 \KV@sp@def\MT@val{###1}%
2876 \MT@ifstreq\MT@val*\def\MT@val{*/*/*/*}\relax
2877 \expandafter\MT@get@font@and@size\MT@val////\@nil
2878 \MT@xdef\MT@MT@permutelist @\@tempb\MT@extra@context}%
2879 {\csname MT@MT@permutelist @name\endcsname}%
2880 <debug>\MT@info@n1{1}{initialising: use list for font \@tempb=\MT@val
2881 <debug> \ifx\MT@extra@context\@empty\else\MessageBreak
2882 <debug> (context: \MT@extra@context)\fi}%
2883 \MT@exp@cs\MT@xaddb
2884 {MT@MT@permutelist @\@tempb\MT@extra@context @sizes}%
2885 {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
2886 }%
2887 }%
2888 }

```

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

```

2889 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2890 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
2891 }

2892 \MT@define@code@key{encoding}{cfg}
2893 \MT@define@code@key{family}{cfg}
2894 \MT@define@code@key{series}{cfg}
2895 \MT@define@code@key{shape}{cfg}
2896 \MT@define@code@key@size{cfg}
2897 \MT@define@code@key@font{cfg}

```

\MT@define@opt@key

```

2898 \def\MT@define@opt@key#1#2{%
2899 \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
2900 \MT@xdef\MT@#1@c@\MT@curr@set@name @#2}{##1}}}%
2901 }

```

The options in the optional first argument.

```

2902 \MT@map@clist@c\MT@features{%

```

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

```

2903 \define@key{MT@#1@c}{name}[] {%
2904   \MT@ifempty{##1}{%
2905     \MT@edefn{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2906   }{%
2907     \MT@edefn{MT@#1@c@name}{##1}%
2908     \MT@ifdefined@nT{MT@#1@c@csname MT@#1@c@name\endcsname}{%
2909       \MT@warning{Redefining \@nameuse{MT@abbr@#1} list ~\@nameuse{MT@#1@c@name}'}%
2910     }%
2911   }%
2912   \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
2913 }%
2914 \MT@define@opt@key{#1}{load}%
2915 \MT@define@opt@key{#1}{factor}%
2916 \MT@define@opt@key{#1}{preset}%
2917 \MT@define@opt@key{#1}{inputenc}%

```

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

```

2918 \define@key{MT@#1@c}{context}[] {\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
2919 }

```

Automatically enable font copying if we find a protrusion or expansion context. After the preamble, check whether font copying is enabled. For older pdfTeX versions, disallow. Also disable for LuaTeX.

```

2920 \MT@requires@pdftex7{
2921   (*lua)
2922   \MT@requires@luatex{
2923     \define@key{MT@ex@c}{context}[] {%
2924       \MT@error{Expansion contexts currently don't work with luatex.\MessageBreak
2925         Ignoring `context' key\on@line}%
2926       {Use pdftex instead.}%
2927     }
2928   }{
2929     (/lua)
2930     \define@key{MT@ex@c}{context}[] {%
2931       \MT@ifempty{#1}\relax{%
2932         \MT@gl@t\MT@copy@font\MT@copy@font@
2933         \def\MT@extra@context{#1}%
2934       }%
2935     }
2936     \MT@addto@setup{%
2937       \define@key{MT@ex@c}{context}[] {%
2938         \ifx\MT@copy@font\MT@copy@font@
2939           \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
2940         \else
2941           \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
2942             Ignoring `context' key\on@line}%
2943           {Either move the settings inside the preamble,\MessageBreak
2944             or load the package with the `copyfonts' option.}%
2945         \fi
2946       }
2947     }

```

Protrusion contexts *may* also work without copying the font, so we don't issue an error but only a warning. The problem is that pdfTeX only allows one set of protrusion factors for a given font within one paragraph (those that are in effect at the end of the paragraph will be in effect for the whole paragraph). When different fonts are loaded – like in the example with the footnote markers – we don't need

to copy the fonts.

```

2948 \define@key{MT@pr@c}{context}[]{%
2949 \MT@ifempty{#1}\relax{%
2950 \MT@glet\MT@copy@font\MT@copy@font@
2951 \def\MT@extra@context{#1}%
2952 }%
2953 }
2954 \MT@addto@setup{%
2955 \define@key{MT@pr@c}{context}[]{%
2956 \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
2957 \ifx\MT@copy@font\MT@copy@font@else
2958 \MT@warning@n{If protrusion contexts don't work as expected,
2959 \MessageBreak load the package with the `copyfonts' option}%
2960 \fi
2961 }
2962 }
2963 \lua{}
2964 }{
2965 \define@key{MT@ex@c}{context}[]{%
2966 \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
2967 or later. Ignoring `context' key\on@line}%
2968 {Upgrade pdftex.}%
2969 }
2970 }

```

\MT@warn@nodim

```

2971 \def\MT@warn@nodim#1{%
2972 \MT@warning{\@tempa' is not a dimension.\MessageBreak
2973 Ignoring it and setting values relative to\MessageBreak #1}%
2974 }

```

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

```

2975 \define@key{MT@pr@c}{unit}[character]{%
2976 \MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
2977 \def\@tempa{#1}%
2978 \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.

```

2979 \MT@ifdimen\@tempa
2980 {\MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
2981 {\MT@warn@nodim{character widths}}%
2982 }%
2983 }

```

Tracking may only be relative to a dimension.

```

2984 \define@key{MT@tr@c}{unit}[1em]{%
2985 \MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
2986 \def\@tempa{#1}%
2987 \MT@ifdimen\@tempa
2988 {\MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
2989 {\MT@warn@nodim{1em}}%
2990 \MT@gdefn{MT@tr@c@MT@curr@set@name @unit}{1em}}%
2991 }

```

Spacing and kerning codes may additionally be relative to space dimensions.

```

2992 \MT@map@clist@n{sp,kn}{%
2993 \define@key{MT@#1@c}{unit}[space]{%
2994 \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
2995 \def\@tempa{##1}%
2996 \MT@ifstreq\@tempa{character}\relax{%
2997 \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne

```



```

2998     \MT@ifstreq\@tempa{space}\relax{%
2999     \MT@ifdimen\@tempa
3000     {\MT@glletenc{MT@#1@c@\MT@curr@set@name @unit}\@tempa}%
3001     {\MT@warn@nodim{width of space}}}%
3002   }%
3003 }%
3004 }%
3005 }

```

The first argument to `\SetExpansion` accepts some more options.

```

3006 \MT@map@clist@n{stretch,shrink,step}%
3007 \define@key{MT@ex@c}{#1}[]{%
3008   \MT@ifempty{##1}\relax{%
3009     \MT@ifint{##1}%

```

A space terminates the number.

```

3010     \MT@gdef@n{MT@ex@c@\MT@curr@set@name @#1}{##1 }%
3011   }%
3012   \MT@warning{%
3013     Value `##1' for option `#1' is not a number.\MessageBreak
3014     Ignoring it}%
3015 }%
3016 }%
3017 }%
3018 }
3019 \define@key{MT@ex@c}{auto}[true]{%
3020   \def\@tempa{#1}%
3021   \csname if\@tempa\endcsname

```

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

```

3022   \MT@requires@pdfTEX4{%
3023     \MT@gdef@n{MT@ex@c@\MT@curr@set@name @auto}{autoexpand}%
3024   }%
3025   \MT@warning{pdfTEX too old for automatic font expansion}%
3026 }
3027 \else
3028   \MT@requires@pdfTEX4{%
3029     \MT@glletenc{MT@ex@c@\MT@curr@set@name @auto}\@empty
3030   }\relax
3031 \fi
3032 }

```

Tracking: Interword spacing and outer kerning. The variant with space in case `\SetTracking` is called inside an argument (e. g., to `\IfFileExists`).

```

3033 \MT@define@opt@key{tr}{spacing}
3034 \MT@define@opt@key{tr}{outerspacing}
3035 \MT@define@opt@key{tr}{outerkerning}

```

Which ligatures should be disabled?

```

3036 \define@key{MT@tr@c}{noligatures}[]%
3037   {\MT@xdef@n{MT@tr@c@\MT@curr@set@name @noligatures}{#1}}
3038 \define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
3039 \define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
3040 \define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{noligatures={#1}}}

```

14.3.6 Character inheritance

`\DeclareCharacterInheritance` This macro may be used in the configuration files to declare characters that should inherit protrusion resp. expansion values from other characters. Thus, there is no need to define all accented characters (e. g., `\'a`, `\'a`, `\^a`, `\~a`, `\"a`, `\r{a}`),

`\k{a}`, `\u{a}`), which will make the configuration files look much nicer and easier to maintain. If a single character of an inheritance list should have a different value, one can simply override it.

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

```
3041 \renewcommand*\DeclareCharacterInheritance[1][]{%
3042   \let\MT@extra@context\empty
3043   \let\MT@extra@inputenc\undefined
3044   \let\MT@inh@feat\empty
3045   \setkeys{MT@inh@}{#1}%
3046   \MT@begin@catcodes
3047   \MT@set@inh@list
3048 }
```

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

```
3049 \def\MT@set@inh@list#1#2{%
3050   \MT@ifempty\MT@inh@feat{%
3051     \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{##2}}}%
3052   }%
3053   \MT@map@clist@c\MT@inh@feat{{%
3054     \KV@esp@def\@tempa{##1}%
3055     \MT@ifempty\@tempa\relax{%
3056       \MT@exp@one@n\MT@declare@char@inh
3057       {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
3058     }%
3059   }}%
3060   }%
3061   \MT@end@catcodes
3062 }
```

The keys for the optional argument.

```
3063 \MT@map@clist@c\MT@features@long{%
3064   \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}%
3065   \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
```

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

```
3066 \def\MT@declare@char@inh#1#2#3{%
3067   \MT@edef@n{MT@#1@inh@name}%
3068   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3069   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3070   \MT@ifdefined@cT\MT@extra@inputenc{%
3071     \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3072   (debug) \MT@info{1}{creating inheritance list \@nameuse{MT@#1@inh@name}}%
3073   \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
3074   \def\MT@permute@list{#1@inh}%
3075   \setkeys{MT@inh@}{#2}%
3076   \MT@permute
3077 }
```

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

```
3078 \define@key{MT@inh@}{encoding}[]{%
3079   \def\MT@val{#1}%
3080   \expandafter\MT@encoding@check\MT@val,\@nil
3081   \MT@get@highlevel{encoding}%
3082   \MT@edef@n{MT@tempencoding1}{\MT@val}%
3083 }
```

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

```
3084 \def\MT@encoding@check#1,#2\@nil{%
```

```

3085 \MT@ifempty{#2}\relax{%
3086   \edef\MT@val{#1}%
3087   \MT@warning{You may only specify one encoding for character\MessageBreak
3088             inheritance lists. Ignoring encoding(s) #2}%
3089 }%
3090 }

```

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

```

3091 \MT@define@code@key{family}{inh}
3092 \MT@define@code@key{series}{inh}
3093 \MT@define@code@key{shape}{inh}
3094 \MT@define@code@key{size}{inh}
3095 \MT@define@code@key{font}{inh}

```

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

```

3096 \def\MT@inh@do#1,{%
3097   \ifx\relax#1\@empty \else
3098     \MT@inh@split #1==\relax
3099     \expandafter\MT@inh@do
3100   \fi
3101 }

```

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

```

3102 \def\MT@inh@split#1=#2=#3\relax{%
3103   \def\@tempa{#1}%
3104   \ifx\@tempa\@empty \else
3105     \MT@get@slot
3106     \ifnum\MT@char > \m@ne
3107       \let\MT@val\MT@char
3108       \MT@map@clist@n{#2}{%
3109         \def\@tempa{##1}%
3110         \ifx\@tempa\@empty \else
3111           \MT@get@slot
3112           \ifnum\MT@char > \m@ne
3113             \MT@exp@cs\MT@xadd{\MT@inh@\MT@listname @\MT@val @}{\{\MT@char\}}%
3114           \fi
3115         \fi
3116       }%
3117   <debug>\MT@info@n1{2}{children of #1 (\MT@val):
3118   <debug> \@nameuse{\MT@inh@\MT@listname @\MT@val @}}%
3119   \fi
3120 \fi
3121 }

```

14.3.7 Permutation

`\MT@permute` Calling `\MT@permute` will define commands for all permutations of the specified font attributes of the form `\MT@<list type>@/<encoding>/<family>/<series>/<shape>/<|*>` to be the expansion of `\MT@<list type>@name`, i. e., the name of the currently defined list.

`\MT@permute@@` Size ranges are held in a separate macro called `\MT@<list type>@/@sizes`, which in turn contains the respective *<list name>*s attached to the ranges.

`\MT@permute@@@`

```

3122 \def\MT@permute{%
3123   \let\MT@cnt@encoding\@ne

```

```

3124 \MT@permute@
      Undefine commands for the next round.
3125 \MT@map@tlist@n{{encoding}}{family}{series}{shape}}\MT@permute@reset
3126 \MT@gl@et\MT@tempsize\@undefined
3127 }
3128 \def\MT@permute@{%
3129 \let\MT@cnt@family\@ne
3130 \MT@permute@@
3131 \MT@increment\MT@cnt@encoding
3132 \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
3133 \MT@permute@
3134 }
3135 \def\MT@permute@@{%
3136 \let\MT@cnt@series\@ne
3137 \MT@permute@@@
3138 \MT@increment\MT@cnt@family
3139 \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
3140 \MT@permute@@
3141 }
3142 \def\MT@permute@@@{%
3143 \let\MT@cnt@shape\@ne
3144 \MT@permute@@@@
3145 \MT@increment\MT@cnt@series
3146 \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
3147 \MT@permute@@@@
3148 }
3149 \def\MT@permute@@@@{%
3150 \MT@permute@@@@@
3151 \MT@increment\MT@cnt@shape
3152 \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
3153 \MT@permute@@@@@
3154 }

```

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

```

3155 \def\MT@permute@@@@@{%
3156 \MT@permute@define{encoding}%
3157 \ifMT@document
3158 \ifx\MT@tempencoding\@empty \else
3159 \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
3160 {\expandafter\expandafter\expandafter\@gobble}%
3161 \fi
3162 \fi
3163 \MT@permute@@@@@@
3164 }

```

\MT@permute@@@@@@

```

3165 \def\MT@permute@@@@@@{%
3166 \MT@permute@define{family}%
3167 \MT@permute@define{series}%
3168 \MT@permute@define{shape}%
3169 \edef\@tempa{\MT@tempencoding
3170 \MT@tempfamily
3171 \MT@tempseries
3172 \MT@tempshape
3173 \MT@ifdefined@c@T\MT@tempsize *}%

```

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

```

3174 \MT@ifstreq\@tempa{////}\relax{%
3175 \ifx\MT@tempencoding\@empty
3176 \MT@warning{%

```

```

3177         You have to specify an encoding for\MessageBreak
3178         \@nameuse{MT@abbr@MT@permutelist} list
3179         ~\@nameuse{MT@MT@permutelist @name}'.\MessageBreak
3180         Ignoring it}%
3181     \else
3182         \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

3183         \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3184         \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3185         }%
3186         \MT@exp@cs\MT@xaddb
3187         {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3188         \MT@tempsize
3189     <debug> \MT@info@n1{1}{initialising: use list for font \@tempa,\MessageBreak
3190     <debug>         sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3191     <debug>         @sizes\endcsname}%
3192     }{%

```

Only one list can apply to a given combination.

```

3193         \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3194         \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3195         ~\@nameuse{MT@MT@permutelist @name}' will override list\MessageBreak
3196         ~\@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3197         for font \@tempa'}%
3198         }%
3199     <debug> \MT@info@n1{1}{initialising: use list for font \@tempa
3200     <debug>         \ifx\MT@extra@context\@empty\else\MessageBreak
3201     <debug>         (context: \MT@extra@context)\fi}%
3202     }%
3203     \MT@xdef@n{MT@MT@permutelist @\@tempa\MT@extra@context}%
3204     {\csname MT@MT@permutelist @name\endcsname}%
3205     \fi
3206     }%
3207 }

```

\MT@permute@define Define the commands.

```

3208 \def\MT@permute@define#1{%
3209     \@tempcnta=\csname MT@cnt@#1\endcsname\relax
3210     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3211     {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3212     {\MT@let@nc{MT@temp#1}\@empty}%
3213 }

```

\MT@permute@reset Reset the commands.

```

3214 \def\MT@permute@reset#1{%
3215     \@tempcnta=\@ne
3216     \MT@loop
3217     \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3218     \advance\@tempcnta\@ne
3219     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3220     \iftrue
3221     \iffalse
3222     \MT@repeat
3223 }

```

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

```

3224 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

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

```
3225 \def\MT@check@rlist@#1#2#3{%
3226   \def\@tempb{#1}%
3227   \def\@tempc{#2}%
3228   \MT@if@false
3229   \MT@exp@cs\MT@map@tlist@c
3230   {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3231   \MT@check@range
3232 }
```

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

```
3233 \def\MT@check@range#1{\expandafter\MT@check@range@ #1}
```

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

```
3234 \def\MT@check@range@#1#2#3{%
3235   \MT@ifdim{#2}=\m@ne{%
3236     \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
3237     \MT@ifdim\@tempb={#1}\MT@if@true\relax
3238   }{%
```

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

```
3239     \MT@ifdim\@tempb>{#1}\relax{%
3240       \MT@ifdim\@tempc>{#1}{%
3241         \MT@if@true
3242         \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3243       }\relax
3244     }%
3245   }%
3246 }{%
3247   \MT@ifdim\@tempc=\m@ne{%
```

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

```
3248     \MT@ifdim\@tempb<{#2}{%
3249       \MT@ifdim\@tempb<{#1}\relax\MT@if@true
3250     }\relax
3251   }{%
```

- Both items are ranges.

```
3252     \MT@ifdim\@tempb<{#2}{%
3253       \MT@ifdim\@tempc>{#1}{%
3254         \MT@if@true
3255         \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3256       }\relax
3257     }\relax
3258   }%
3259 }%
3260 \ifMT@if@
3261   \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
3262     \@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
3263     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.

```
3264   \expandafter\MT@tlist@break
3265 \fi
3266 }
```

14.4 Package options

14.4.1 Declaring the options

```

\ifMT@opt@expansion      Keep track of whether the user explicitly set these options.
\ifMT@opt@auto 3267 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 3268 \newif\ifMT@opt@auto
                 3269 \newif\ifMT@opt@DVI

\MT@optwarn@admissible   Some warnings.
3270 \def\MT@optwarn@admissible#1#2{%
3271   \MT@warning@n1{`#1' is not an admissible value for option\MessageBreak
3272     `#2'. Assuming `false'}%
3273 }

\MT@optwarn@nan
3274 </package>
3275 <plain>\MT@requires@latex1{
3276 \def\MT@optwarn@nan#1#2{%
3277   \MT@warning@n1{Value `#1' for option `#2' is not a\MessageBreak number.
3278     Using default value of \number\@nameuse{MT@#2@default}}%
3279 }
3280 <plain>\relax
3281 <*package>

\MT@opt@def@set
3282 \def\MT@opt@def@set#1{%
3283   \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}}{%
3284     \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
3285   }{%
3286     \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
3287     \MT@warning@n1{The #1 set `MT@val' is undeclared.\MessageBreak
3288       Using set ` \@nameuse{MT@\@tempb @setname}' instead}%
3289   }%
3290 }

expansion and protrusion may be true, false, compatibility, nocompatibility
and/or a <set name>.
3291 \MT@map@clist@n{protrusion,expansion}{%
3292   \define@key{MT}{#1}[true]{%
3293     \csname MT@opt@#1true\endcsname
3294     \MT@map@clist@n{##1}{%
3295       \KV@esp@def\MT@val{###1}%
3296       \MT@ifempty\MT@val\relax{%
3297         \csname MT@#1true\endcsname
3298         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3299         \MT@ifstreq\MT@val{true}\relax
3300       }{%
3301         \MT@ifstreq\MT@val{false}{%
3302           \csname MT@#1false\endcsname
3303         }{%
3304           \MT@ifstreq\MT@val{compatibility}{%
3305             \MT@let@nc{MT@\@tempb @level}\@ne
3306           }{%
3307             \MT@ifstreq\MT@val{nocompatibility}{%
3308               \MT@let@nc{MT@\@tempb @level}\tw@
3309             }{%
3310               \MT@opt@def@set{#1}%
3311             }%

```

```

3312         }%
3313     }%
3314 }%
3315 }%
3316 }%
3317 }%
3318 }

```

activate is a shortcut for protrusion and expansion.

```

3319 \define@key{MT}{activate}[true]{%
3320     \setkeys{MT}{protrusion={#1}}%
3321     \setkeys{MT}{expansion={#1}}%
3322 }

```

spacing, kerning and tracking do not have a compatibility level.

```

3323 \MT@map@clist@n{spacing,kerning,tracking}{%
3324     \define@key{MT}{#1}[true]{%
3325         \MT@map@clist@n{##1}{%
3326             \KV@sp@def\MT@val{###1}%
3327             \MT@ifempty\MT@val\relax{%
3328                 \csname MT@#1true\endcsname
3329                 \MT@ifstreq\MT@val{true}\relax
3330             }%
3331             \MT@ifstreq\MT@val{false}{%
3332                 \csname MT@#1false\endcsname
3333             }%
3334             \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3335             \MT@opt@def@set{#1}%
3336         }%
3337     }%
3338 }%
3339 }%
3340 }%
3341 }

```

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

```

3342 \def\MT@def@bool@opt#1#2{%
3343     \define@key{MT}{#1}[true]{%
3344         \def\@tempa{#1}%
3345         \MT@ifstreq\@tempa{true}\relax{%
3346             \MT@ifstreq\@tempa{false}\relax{%
3347                 \MT@optwarn@admissible{##1}{#1}%
3348             }%
3349         }%
3350     }%
3351     #2%
3352 }%
3353 }

```

Boolean options that only set the switch.

```

3354 \MT@map@clist@n{draft,selected,babel}{%
3355     \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
3356 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotru}

```

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

```

3357 \MT@def@bool@opt{DVIoutput}{%
3358     \csname if\@tempa\endcsname
3359     \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
3360     \pdfoutput\z@
3361 }%

```



```

3362 \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
3363 \pdfoutput\@ne
3364 \fi
3365 }

```

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

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

```

3366 \MT@def@bool@opt{defersetup}{%
3367 \csname if\@tempa\endcsname \else
3368 \AtEndOfPackage{%
3369 \MT@setup@
3370 \let\MT@setup@\empty
3371 \let\MT@addto@setup\@firstofone
3372 }%
3373 \fi
3374 }

```

`copyfonts` will copy all fonts before setting them up. This allows protrusion and expansion with different parameters. This options is also *undocumented* in the hope that we can always find out automatically whether it's required.

```

3375 \MT@requires@pdftex7{
3376 (*lua)
3377 \MT@requires@luatex{
3378 \MT@def@bool@opt{copyfonts}{%
3379 \csname if\@tempa\endcsname
3380 \MT@error{The `copyfonts' option doesn't work with luatex}
3381 {Use pdftex instead.}%
3382 \fi
3383 }
3384 }{
3385 (/lua)
3386 \MT@def@bool@opt{copyfonts}{%
3387 \csname if\@tempa\endcsname
3388 \MT@gl@et\MT@copy@font\MT@copy@font@
3389 \else
3390 \MT@gl@et\MT@copy@font\relax
3391 \fi
3392 }
3393 (lua) }
3394 }{
3395 \MT@def@bool@opt{copyfonts}{%
3396 \csname if\@tempa\endcsname
3397 \MT@error{The pdftex version you are using is too old\MessageBreak
3398 to use the `copyfonts' option}{Upgrade pdftex.}%
3399 \fi
3400 }
3401 }

```

`final` is the opposite to `draft`.

```

3402 \MT@def@bool@opt{final}{%
3403 \csname if\@tempa\endcsname
3404 \MT@draftfalse
3405 \else
3406 \MT@drafttrue
3407 \fi
3408 }

```

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

```
3409 \define@key{MT}{verbose}[true]{%
3410   \let\MT@vinfo\MT@info@n1
3411   \def\@tempa{#1}%
3412   \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
3413   \MT@ifstreq\@tempa{errors}{%
3414     \let\MT@warning \MT@warn@err
3415     \let\MT@warning@n1\MT@warn@err
3416   }{%
3417     \let\MT@vinfo\@gobble
```

Cast warnings to the winds.

```
3418   \MT@ifstreq\@tempa{silent}{%
3419     \let\MT@warning \MT@info
3420     \let\MT@warning@n1\MT@info@n1
3421   }{%
3422     \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
3423   }%
3424 }%
3425 }%
3426 }
```

Options with numerical keys: factor, stretch, shrink, step, letterspace.

```
3427 </package>
3428 <plain>\MT@requires@latex1{
3429 \MT@map@clist@n{%
3430 <package> stretch,shrink,step,%
3431 letterspace}{%
3432 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
3433 \def\@tempa{##1 }%
```

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

```
3434 \MT@ifint\@tempa
3435 {\MT@edef@n{MT@#1}{\@tempa}}%
3436 {\MT@optwarn@nan{##1}{#1}}%
3437 }%
3438 }
3439 <plain>\relax
3440 <*package>
```

`factor` will define the protrusion factor only.

```
3441 \define@key{MT}{factor}[\MT@factor@default]{%
3442 \def\@tempa{#1 }%
3443 \MT@ifint\@tempa
3444 {\edef\MT@pr@factor{\@tempa}}
3445 {\MT@optwarn@nan{#1}{factor}}%
3446 }
```

Unit for protrusion codes.

```
3447 \define@key{MT}{unit}[character]{%
3448 \def\@tempa{#1}%
3449 \MT@ifstreq\@tempa{character}\relax{%
3450   \MT@ifdimen\@tempa
3451   {\let\MT@pr@unit\@tempa}%
3452   {\MT@warning@n1{~\@tempa' is not a dimension.\MessageBreak
3453     Ignoring it and setting values relative to\MessageBreak
3454     character widths}}%
3455 }%
3456 }
```

14.4.2 Reading the configuration file

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

```
3457 \MT@protrusiontrue
3458 \ifnum\pdfoutput<\@ne \else
```

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

```
3459 \MT@requires@pdftex4{
3460   \MT@expansiontrue
3461   \MT@autottrue
3462 } \relax
3463 \fi
```

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

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

```
3464 \define@key{MT}{config}[] {\relax}
3465 \def\MT@get@config#1config=#2,#3\@nil{%
3466   \MT@ifempty{#2}%
3467   {\def\MT@config@file{\MT@MT.cfg}}%
3468   {\def\MT@config@file{#2.cfg}}%
3469 }
3470 \expandafter\expandafter\expandafter\MT@get@config
3471 \curname opt@ \currname. \currxt \endcurname, config=, \@nil
```

Load the file.

```
3472 \IfFileExists{\MT@config@file}{%
3473   \MT@info@nl{Loading configuration file \MT@config@file}%
3474   \MT@begin@catcodes
3475   \let\MT@begin@catcodes\relax
3476   \let\MT@end@catcodes\relax
3477   \let\MT@curr@file\MT@config@file
3478   \input{\MT@config@file}%
3479   \endgroup
3480 }{\MT@warning@nl{%
3481   Could not find configuration file ` \MT@config@file'!\MessageBreak
3482   This will almost certainly cause undesired results.\MessageBreak
3483   Please fix your installation}%
3484 }
```

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

```
3485 \MT@map@clist@ \MT@features{%
3486   \MT@ifdefined@n@TF{MT@default@#1@set} \relax
3487   {\MT@gdef@n{MT@default@#1@set}{all}}%
3488 }
```

`\MT@check@active@set` We have to make sure that font sets are active. If the user didn't activate any, we use those sets declared by `\DeclareMicrotypeSetDefault` (this is done at the end of the preamble).

```
3489 \def\MT@check@active@set#1{%
3490   \MT@ifdefined@n@TF{MT@#1@setname}{%
3491     \MT@info@nl{Using \@nameuse{MT@abbr@#1} set ` \@nameuse{MT@#1@setname}' }%
3492   }%
```

```

3493 \MT@glet@nn{MT@#1@setname}{MT@default@#1@set}%
3494 \MT@info@n1{Using default \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}'}%
3495 }%
3496 }

```

14.4.3 Hook for other packages

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

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

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

```

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

```

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

```

3497 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
3498   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3499   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
3500 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook

```

14.4.4 Changing options later

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

```

3501 \def\microtypesetup{\setkeys{MT}}
3502 \MT@addto@setup{\def\microtypesetup{\setkeys{MTX}}}
3503 \def\MT@define@optionX#1#2{%
3504   \define@key{MTX}{#1}[true]{%
3505     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3506     \MT@map@clist@n{##1}{%
3507       \KV@sp@def\MT@val{###1}%
3508       \MT@ifempty\MT@val\relax{%
3509         \@tempcnta=\m@ne
3510         \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.

```

3511 \MT@checksetup{#1}{%
3512 \@tempcnta=\csname MT@\@tempb @level\endcsname
3513 \MT@info{Enabling #1
3514 (level \number\csname MT@\@tempb @level\endcsname)}%
3515 }%
3516 }{%
3517 \MT@ifstreq\MT@val{false}{%
3518 \@tempcnta=\z@
3519 \MT@info{Disabling #1}%
3520 }{%
3521 \MT@ifstreq\MT@val{compatibility}{%
3522 \MT@checksetup{#1}{%
3523 \@tempcnta=\@ne
3524 \MT@let@nc{MT@\@tempb @level}\@ne
3525 \MT@info{Setting #1 to level 1}%
3526 }%
3527 }{%
3528 \MT@ifstreq\MT@val{nocompatibility}{%
3529 \MT@checksetup{#1}{%
3530 \@tempcnta=\tw@
3531 \MT@let@nc{MT@\@tempb @level}\tw@
3532 \MT@info{Setting #1 to level 2}%
3533 }%
3534 }{\MT@error{Value `MT@val' for key `#1' not recognised}
3535 {Use any of `true', `false', `compatibility' or
3536 `nocompatibility'.}%
3537 }%
3538 }%
3539 }%
3540 }%
3541 \ifnum\@tempcnta>\m@ne
3542 #2\@tempcnta\relax
3543 \fi
3544 }%
3545 }%
3546 }%
3547 }

```

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

```

3548 \def\MT@checksetup#1{%
3549 \csname ifMT@#1\endcsname
3550 \expandafter\@firstofone
3551 \else
3552 \MT@error{You cannot enable #1 if it was disabled\MessageBreak
3553 in the package options}{Load microtype with #1 enabled.}%
3554 \expandafter\@gobble
3555 \fi
3556 }

3557 \MT@define@optionX{protrusion}\pdfprotrudechars
3558 \MT@define@optionX{expansion}\pdfadjustspacing

```

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

```

3559 \MT@requires@pdftex6{
3560 \Lua \MT@requires@luatex\@firstofone{
3561 \def\MT@define@optionX@#1#2{%
3562 \define@key{MTX}{#1}[true]{%
3563 \MT@map@clist@n{##1}{%

```

```

3564 \KV@sp@def\MT@val{###1}%
3565 \MT@ifempty\MT@val\relax{%
3566 \@tempcnta=\m@ne
3567 \MT@ifstreq\MT@val{true}{%
3568 \MT@checksetup{#1}%
3569 \@tempcnta=\@ne
3570 \MT@info{Enabling #1}%
3571 }%
3572 }{%
3573 \MT@ifstreq\MT@val{false}{%
3574 \@tempcnta=\z@
3575 \MT@info{Disabling #1}%
3576 }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
3577 {Use either `true' or `false'}}%
3578 }%
3579 }%
3580 \ifnum\@tempcnta>\m@ne
3581 #2\relax
3582 \fi
3583 }%
3584 }%
3585 }%
3586 }

3587 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\relax
3588 \else\let\MT@tracking\MT@tracking@fi}
3589 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
3590 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3591 \pdfappendkern \@tempcnta}
3592 \gobble
3593 lua }
3594 } \@firstofone

```

Disable for older pdf_{TEX} versions and for Lua_{TEX}.

```

3595 {\define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
3596 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
3597 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
3598 }
3599 \define@key{MTX}{activate}[true]{%
3600 \setkeys{MTX}{protrusion={#1}}%
3601 \setkeys{MTX}{expansion={#1}}%
3602 }

```

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

```

3603 \let\MT@saved@setupfont\MT@setupfont
3604 \define@key{MTX}{disable}[]{%
3605 \MT@info{Inactivate `~\MT@MT' package}%
3606 \let\MT@setupfont\relax
3607 }
3608 \define@key{MTX}{enable}[]{%
3609 \MT@info{Reactivate `~\MT@MT' package}%
3610 \let\MT@setupfont\MT@saved@setupfont
3611 }
3612 /package

```

14.4.5 Processing the options

`\MT@ProcessOptionsWithKV` Parse options.

```

3613 plain\MT@requires@latex1{
3614 \def\MT@ProcessOptionsWithKV#1{%

```

```

3615 \let\@tempc\relax
3616 \let\MT@temp\@empty
3617 plain \MT@requires@latex2{
3618   \MT@map@clist@{\@classoptionslist}%
3619   \def\CurrentOption{##1}%
3620   \MT@ifdefined@n@T{KV@#1@}\expandafter\MT@getkey\CurrentOption=\@nil}{%
3621     \edef\MT@temp{\MT@temp,\CurrentOption,}%
3622     \expandtwoargs\@removeelement\CurrentOption
3623     \@unusedoptionlist\@unusedoptionlist
3624   }%
3625 }%
3626 \edef\MT@temp{\noexpand\setkeys{#1}%
3627   {\MT@temp\@optionlist{\@currname.\@currentext}}}%

```

plain can handle package options.

```

3628 *plain
3629   {\edef\MT@temp{\noexpand\setkeys{#1}%
3630     {\csname usepkg@options@usepkg@pkg\endcsname}}}
3631 /plain
3632 \MT@temp
3633 \MT@clear@options
3634 }

```

\MT@getkey For key=val in class options.

```

3635 \def\MT@getkey#1=#2\@nil{#1}
3636 \MT@ProcessOptionsWithKV{MT}
3637 plain \relax
3638 *package

```

Now we can take the appropriate actions. We also tell the log file which options the user has chosen (in case it's interested).

```

3639 \MT@addto@setup{%
3640 \ifMT@draft

```

We disable most of what we've just defined in the 3640 lines above if we are running in draft mode.

```

3641 \MT@warning@nl{'draft' option active.\MessageBreak
3642   Disabling all micro-typographic extensions.\MessageBreak
3643   This might lead to different line and page breaks}
3644 \let\MT@setupfont\relax
3645 \renewcommand*\LoadMicrotypeFile[1]{}
3646 \renewcommand*\microtypesetup[1]{}
3647 \renewcommand*\microtypecontext[1]{}
3648 \renewcommand*\lsstyle{}
3649 \else

```

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

```

3650 \ifnum\pdfoutput<\@ne
3651   \ifMT@opt@expansion \else
3652     \MT@expansionfalse
3653   \fi
3654 \fi

```

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

Some packages depend on the value of \pdfoutput and will get confused if it is changed after they have been loaded. These packages are, among others: color, graphics, hyperref, crop, contour, pstricks and, as a matter of course, ifpdf.

Instead of testing for each package (that's not our job), we only say that it was microtype that changed it. This must be sufficient!

```
3655 \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3656 \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
```

Fix the font sets.

```
3657 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
```

Protrusion.

```
3658 \ifMT@protrusion
3659 \edef\MT@active@features{\MT@active@features,pr}
3660 \pdfprotrudechars\MT@pr@level
3661 \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
3662 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3663 factor: \number\MT@pr@factor\fi
3664 \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}
3665 \MT@check@active@set{pr}
3666 \else
3667 \let\MT@protrusion\relax
3668 \MT@info@nl{No character protrusion}
3669 \fi
```

Expansion.

```
3670 \ifMT@expansion
```

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

```
3671 \ifnum\MT@stretch=\m@ne
3672 \let\MT@stretch\MT@stretch@default
3673 \fi
```

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

```
3674 \ifnum\MT@shrink=\m@ne
3675 \let\MT@shrink\MT@stretch
3676 \fi
```

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

```
3677 \ifnum\MT@step=\m@ne
3678 \ifnum\MT@stretch>\MT@shrink
3679 \ifnum\MT@shrink=\z@
3680 \@tempcnta=\MT@stretch
3681 \else
3682 \@tempcnta=\MT@shrink
3683 \fi
3684 \else
3685 \ifnum\MT@stretch=\z@
3686 \@tempcnta=\MT@shrink
3687 \else
3688 \@tempcnta=\MT@stretch
3689 \fi
3690 \fi
3691 \divide\@tempcnta 5\relax
3692 \else
3693 \@tempcnta=\MT@step
3694 \ifnum\@tempcnta=\z@
3695 \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
3696 Setting it to one}
3697 \fi
3698 \fi
3699 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
```


3700 \edef\MT@step{\number\@tempcnta\space}
\MT@auto Automatic expansion of the font? This new feature of pdf \TeX 1.20 makes the *l^AT_EX* programme really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdf \TeX).

3701 \let\MT@auto\empty
3702 \ifMT@auto
3703 \MT@requires@pdftex4{%

We turn off automatic expansion if output mode is DVI.

3704 \ifnum\pdfoutput<\@ne
3705 \ifMT@opt@auto
3706 \MT@error{%
3707 Automatic font expansion only works for PDF output.\MessageBreak
3708 However, you are creating a DVI file}
3709 {If you have created expanded fonts instances, remove ‘auto’ from
3710 \MessageBreak the package options. Otherwise, you have to switch
3711 off expansion\MessageBreak completely.}
3712 \fi
3713 \MT@autofalse
3714 \else
3715 \def\MT@auto{autoexpand}
3716 \fi

Also, if pdf \TeX is too old.

3717 }{%
3718 \MT@error{%
3719 The pdftex version you are using is too old for\MessageBreak
3720 automatic font expansion}%
3721 {If you have created expanded fonts instances, remove ‘auto’ from\MessageBreak
3722 the package options. Otherwise, you have to switch off expansion\MessageBreak
3723 completely, or upgrade pdftex to version 1.20 or newer.}
3724 \MT@autofalse
3725 \def\MT@auto{1000 }
3726 }
3727 \else

No automatic expansion.

3728 \MT@requires@pdftex4\relax{
3729 \def\MT@auto{1000 }
3730 }
3731 \fi

Choose the appropriate macro for selected expansion.

3732 \ifMT@selected
3733 \let\MT@set@ex@codes\MT@set@ex@codes@s
3734 \else
3735 \let\MT@set@ex@codes\MT@set@ex@codes@n
3736 \fi

Filter out stretch=0, shrink=0, since it would result in a pdf \TeX error.

3737 \ifnum\MT@stretch=\z@
3738 \ifnum\MT@shrink=\z@
3739 \MT@warning@n1{%
3740 Both the stretch and shrink limit are set to zero.\MessageBreak
3741 Disabling font expansion}
3742 \MT@expansionfalse
3743 \fi
3744 \fi
3745 \fi
3746 \ifMT@expansion
3747 \edef\MT@active@features{\MT@active@features,ex}%

```

3748 \pdfadjustspacing\MT@ex@level
3749 \MT@info@n1{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
3750 (level \number\MT@ex@level),\MessageBreak
3751 stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3752 step: \number\MT@step, \ifMT@selected\else non-\fi selected%
3753 \ifx\MT@copy@font\relax\else .\MessageBreak
3754 Using font copies for contexts\fi}

```

\MT@check@step Check whether stretch and shrink are multiples of step.

```

3755 \def\MT@check@step#1{%
3756 \tempcnta=\csname MT@#1\endcsname
3757 \divide\tempcnta \MT@step
3758 \multiply\tempcnta \MT@step
3759 \ifnum\tempcnta=\csname MT@#1\endcsname\else
3760 \MT@warning@n1{The #1 amount is not a multiple of step.\MessageBreak
3761 The effective maximum #1 is \the\tempcnta\space
3762 (step \number\MT@step)}
3763 \fi
3764 }
3765 \MT@check@step{stretch}
3766 \MT@check@step{shrink}
3767 \MT@check@active@set{ex}

```

Inside \showhyphens, font expansion should be disabled.

```

3768 \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3769 \color@begingroup\everypar{}\parfillskip\z@skip
3770 \hsize\maxdimen\normal font\pretolerance\m@ne\tolerance\m@ne
3771 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

```

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

```

3772 \gdef\showhyphens#1{\setbox0\vbox{%
3773 \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3774 \hsize\maxdimen\normal font\pretolerance\m@ne\tolerance\m@ne
3775 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

3776 \else
3777 \let\MT@expansion\relax
3778 \MT@info@n1{No font expansion}
3779 \fi
3780 }
3781 \MT@requires@pdftex6{
3782 \*lua
3783 \def\MT@warn@lua#1{%
3784 \MT@error{The '~#1' feature doesn't currently work\MessageBreak with luatex}
3785 {Use pdftex instead.}%
3786 \MT@let@nc{MT@#1}\relax
3787 }
3788 \*lua
3789 \*package
3790 \MT@addto@setup{%
3791 \*package

```

Tracking, spacing and kerning.

```

3792 \ifMT@tracking
3793 \*lua \MT@requires@luatex{\MT@warn@lua{tracking}}{
3794 \edef\MT@active@features{\MT@active@features,tr}
3795 \MT@info@n1{Tracking enabled}
3796 \MT@check@active@set{tr}

```

Enable protrusion for compensation at the line edges.

```

3797 \ifMT@protrusion\else\pdfprotrudechars\@ne\fi
3798 \*lua }
3799 \else

```

```

3800     \let\MT@tracking\relax
3801     \MT@info@nl{No tracking}
3802   \fi
3803   \ifMT@spacing
3804     \lua \MT@requires@luatex{\MT@warn@lua{spacing}}{
3805       \edef\MT@active@features{\MT@active@features,sp}
3806       \pdfadjustinterwordglue\@ne
3807       \MT@info@nl{Adjustment of interword spacing enabled}
3808       \MT@check@active@set{sp}
3809     }
3810   \else
3811     \let\MT@spacing\relax
3812     \MT@info@nl{No adjustment of interword spacing}
3813   \fi
3814   \ifMT@kerning
3815     \lua \MT@requires@luatex{\MT@warn@lua{kerning}}{
3816       \edef\MT@active@features{\MT@active@features,kn}
3817       \pdfprependkern\@ne
3818       \pdfappendkern\@ne
3819       \MT@info@nl{Adjustment of character kerning enabled}
3820       \MT@check@active@set{kn}
3821     }
3822   \else
3823     \let\MT@kerning\relax
3824     \MT@info@nl{No adjustment of character kerning}
3825   \fi
3826 \end{package}

```

\MT@warn@tracking@DVI We issue a warning, when letterspacing in DVI mode, since it will probably not work. We also switch on protrusion if it isn't already, to compensate for the letterspacing kerns.

```

3827   \ifnum\pdfoutput<\@ne
3828     \def\MT@warn@tracking@DVI{%
3829       \MT@warning@nl{%
3830         You are using tracking/letterspacing in DVI mode.\MessageBreak
3831         This will probably not work, unless the post-\MessageBreak
3832         processing program (dvips, dvi2pdf(x), ...) is\MessageBreak
3833         able to create the virtual fonts on the fly}%
3834       \MT@glet\MT@warn@tracking@DVI\relax
3835     }
3836   \else
3837     \def\MT@warn@tracking@DVI{%
3838       \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
3839       \MT@glet\MT@warn@tracking@DVI\relax
3840     }
3841   \fi

3842   \ifnum\MT@letterspace=\m@ne
3843     \let\MT@letterspace\MT@letterspace@default
3844   \else
3845     \MT@ls@too@large\MT@letterspace
3846   \fi
3847 }

```

If pdf_T_EX is too old, we disable tracking, spacing and kerning, and throw an error message.

```

3848 \end{package}
3849 }{
3850   \MT@addto@setup{%
3851     \ifMT@tracking
3852       \MT@error{Tracking only works with pdftex version 1.40\MessageBreak
3853         or newer. Switching it off}{Upgrade pdftex.}%

```

```

3854 \else
3855 \MT@info@n1{No tracking (pdfTeX too old)}
3856 \fi
3857 \ifMT@spacing
3858 \MT@error{Adjustment of interword spacing only works with\MessageBreak
3859 pdfTeX version 1.40 or newer. Switching it off}{Upgrade pdfTeX.}%
3860 \else
3861 \MT@info@n1{No adjustment of interword spacing (pdfTeX too old)}
3862 \fi
3863 \ifMT@kerning
3864 \MT@error{Character kerning only works with\MessageBreak
3865 pdfTeX version 1.40 or newer. Switching it off}{Upgrade pdfTeX.}%
3866 \else
3867 \MT@info@n1{No adjustment of character kerning (pdfTeX too old)}
3868 \fi
3869 }
3870 }

```

Warning if `\nonfrenchspacing` is active, since space factors will be ignored with `\pdfadjustinterwordglue > 0`. Why 1500? Because some packages redefine `\frenchspacing`. See the c.t.t thread ‘`\frenchspacing` with AMS packages and babel’, started by Philipp Lehman on 16 August 2005: [ddtbaj\\$rob\\$1@online.de](mailto:ddtbajrob1@online.de).

```

3871 \MT@requires@pdfTeX6{
3872 \AtBeginDocument{%
3873 \ifMT@spacing
3874 \ifMT@babel \else
3875 \ifnum\sfcode`. > 1500
3876 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
3877 \MT@warning@n1{%
3878 \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
3879 interword spacing will disable it. You might want\MessageBreak
3880 to add \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
3881 to your preamble}%
3882 }%
3883 \fi
3884 \fi
3885 \fi
3886 }
3887 }\relax

```

`\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```

3888 \MT@requires@pdfTeX5{
3889 \MT@addto@setup{%
3890 \ifMT@noligatures \else
3891 \let\MT@noligatures\relax
3892 \fi
3893 }
3894 }\relax

```

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

```

3895 \MT@addto@setup{%
3896 \ifx\MT@active@features\@empty \else
3897 \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3898 \fi
3899 \MT@documenttrue
3900 }

```

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

```

3901 \def\MT@set@babel@context#1{%

```

```

3902 \MT@ifdefined@n@TF{MT@babel@#1}{%
3903 \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
3904 \expandafter\MT@exp@one@n\expandafter\microtypecontext
3905 \csname MT@babel@#1\endcsname
3906 }{%
3907 \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3908 }%
3909 }

```

\MT@shorthandoff Active characters can only be switched off if babel isn't loaded after microtype.

```

3910 \ifpackageloaded{babel}{
3911 \def\MT@shorthandoff#1#2{%
3912 \MT@info@n1{Switching off #1 babel's active characters (#2)}%
3913 \shorthandoff{#2}}
3914 }{
3915 \def\MT@shorthandoff#1#2{%
3916 \MT@error{You must load `babel' before `~\MT@MT'}
3917 {Otherwise, `~\MT@MT' cannot switch off #1 babel's\MessageBreak
3918 active characters.}}
3919 }

```

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

```

3920 \MT@addto@setup{%
3921 \ifMT@babel
3922 \ifpackageloaded{babel}{%
3923 \MT@info@n1{Redefining babel's language switching commands}
3924 \let\MT@orig@select@language\select@language
3925 \def\select@language#1{%
3926 \MT@orig@select@language{#1}%
3927 \MT@set@babel@context{#1}%
3928 }
3929 \let\MT@orig@foreign@language\foreign@language
3930 \def\foreign@language#1{%
3931 \MT@orig@foreign@language{#1}%
3932 \MT@set@babel@context{#1}%
3933 }
3934 \ifMT@kerning

```

Disable French babel's active characters.

```

3935 \MT@if@false
3936 \MT@with@babel@and@T{french} \MT@if@true
3937 \MT@with@babel@and@T{frenchb} \MT@if@true
3938 \MT@with@babel@and@T{français} \MT@if@true
3939 \MT@with@babel@and@T{canadien} \MT@if@true
3940 \MT@with@babel@and@T{acadian} \MT@if@true
3941 \ifMT@if@MT@shorthandoff{French}{:;!}\fi

```

Disable Turkish babel's active characters.

```

3942 \MT@if@false
3943 \MT@with@babel@and@T{turkish} \MT@if@true
3944 \ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi
3945 \fi

```

In case babel was loaded before microtype:

```

3946 \MT@set@babel@context\languagename
3947 }{%
3948 \MT@warning@n1{You did not load the babel package.\MessageBreak
3949 The `babel' option won't have any effect}
3950 }
3951 \fi
3952 }

```

Now we close the \fi from \ifMT@draft.

```

3953 \MT@addto@setup\fi
      Set up the current font, most likely the normal font. This has to come after all of
      the setup (including anything from the preamble) has been dealt with.
3954 \AtBeginDocument\selectfont
\MT@curr@file This is the current file (hopefully with the correct extension).
3955 \edef\MT@curr@file{\jobname.tex}
      Restore catcodes.
3956 </package>
3957 \MT@restore@catcodes
      That was that.
3958 </package|letterspace>

```

15 Configuration files

Let's now write the font configuration files.

```

3959 <*config>
3960

```

15.1 Font sets

We first declare some sets in the main configuration file.

```

3961 <*m-t>
3962 %%% -----
3963 %%% FONT SETS
3964
3965 \DeclareMicrotypeSet{all}
3966 { }
3967
3968 \DeclareMicrotypeSet{allmath}
3969 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
3970
3971 \DeclareMicrotypeSet{alltext}
3972 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
3973
3974 \DeclareMicrotypeSet{bascmath}
3975 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
3976   family   = {rm*,sf*},
3977   series    = {md*},
3978   size      = {normalsize,footnotesize,small,large}
3979 }
3980
3981 \DeclareMicrotypeSet{basictext}
3982 { encoding = {OT1,T1,LY1,OT4,QX,T5},
3983   family   = {rm*,sf*},
3984   series    = {md*},
3985   size      = {normalsize,footnotesize,small,large}
3986 }
3987
3988 \DeclareMicrotypeSet{smallcaps}
3989 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3990   shape     = {sc}
3991 }
3992
3993 \DeclareMicrotypeSet{footnotesize}

```

```

3994 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
3995   size      = {-small}
3996 }
3997
3998 \DeclareMicrotypeSet{scriptsize}
3999 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
4000   size      = {-footnotesize}
4001 }
4002
4003 \DeclareMicrotypeSet{normal font}
4004 { font = */*/*/*/* }
4005

```

The default sets.

```

4006 %%% -----
4007 %%% DEFAULT SETS
4008
4009 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4010 \DeclareMicrotypeSetDefault[expansion]{basictext}
4011 \DeclareMicrotypeSetDefault[spacing]{basictext}
4012 \DeclareMicrotypeSetDefault[kerning]{alltext}
4013 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
4014

```

15.2 Font variants and aliases

```

4015 %%% -----
4016 %%% FONT VARIANTS AND ALIASES
4017

```

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals):

```

4018 \DeclareMicrotypeVariants{x,j,w,a,d,0,1}
4019

```

Other candidates: 2 (proportional digits), e (engraved), f (Fraktur), g (small text), h (shadow), l (outline), n (informal), p (ornaments), r (roman), s (sans serif), t (typewriter). I've omitted them since they seem hardly be used and/or they are actually more than a variant, i. e., they shouldn't share a file.

Fonts that are 'the same': 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.

```

4020 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
4021 \DeclareMicrotypeAlias{aer}{cmr} % ae
4022 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
4023 \DeclareMicrotypeAlias{cmor}{cmr} % eco
4024 \DeclareMicrotypeAlias{hfor}{cmr} % hfoldsty

```

The packages `pxfonts` and `txfonts` inherit Palatino and Times settings respectively, also the \TeX Gyre fonts `Pagella` and `Termes` (formerly: `qfonts`).

```

4025 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
4026 \DeclareMicrotypeAlias{qpl}{ppl} % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

```

The 'FPL Neu' fonts, a 're-implementation' of Palatino.

```

4027 \DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
4028 \DeclareMicrotypeAlias{fp9j}{pplj} % "
4029 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
4030 \DeclareMicrotypeAlias{qtm}{ptm} % TeX Gyre Termes (formerly: qfonts/QuasiTimes)

```

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

The eulervm package virtually extends the Euler fonts.

```
4031 \DeclareMicrotypeAlias{eur}{eur} % Euler VM
4032 \DeclareMicrotypeAlias{zeus}{eus} % "
```

MicroPress's Charter version (chmath).

```
4033 \DeclareMicrotypeAlias{chr}{bch} % CH Math
```

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

```
4034 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
4035 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
4036 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

Euro symbol fonts, to save some files.

```
4037 \DeclareMicrotypeAlias{zpeu}{zpeu} % Adobe Euro sans -> serif
4038 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
4039 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
4040
```

15.3 Interaction with babel

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

```
4041 %%% -----
4042 %%% INTERACTION WITH THE `babel' PACKAGE
4043
4044 \DeclareMicrotypeBabelHook
4045 {english,UKenglish,british,USenglish,american}
4046 {kerning=, spacing=nonfrench}
4047
4048 \DeclareMicrotypeBabelHook
4049 {french,français,acadian,canadien}
4050 {kerning=french, spacing=}
4051
4052 \DeclareMicrotypeBabelHook
4053 {turkish}
4054 {kerning=turkish, spacing=}
4055
```

15.4 Note on admissible characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```
\ : \textbackslash
{ : \textbraceleft
} : \textbraceright
^ : \textasciicircum
% : \%
# : \#
```

Comma and equal sign must be guarded with braces ('{,}', '{=}') to keep keyval happy.

Ligatures and `\mathchardef` symbols have to be specified numerically. Of course, numerical identification is possible in any other case, too.

15.5 Character inheritance

```

4056 </m-t>
4057 <*m-t |zpeu|mv<
4058 %%%
4059 %%% CHARACTER INHERITANCE
4060
4061 </m-t |zpeu|mv<
4062 <*m-t>

```

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

```
4063 \DeclareCharacterInheritance
4064 { encoding = OT1 }
4065 { f = {011}, % ff
4066   i = {\i},
4067   j = {\j},
4068   O = {\O},
4069   o = {\o}
4070 }
4071
```

Candidates here: 028 ('fi'), 029 ('fl'), 030 ('ffi'), 031 ('ffl'), 156 ('IJ' ligature, since L^AT_EX 2005/12/01 accessible as \IJ), 188 ('ij', \ij), Æ, æ, Œ, œ.

```

4072 \DeclareCharacterInheritance
4073 { encoding = T1 }
4074 { A = {\^A,\^A,\^A,\^-A,\^A,\r A,\k A,\u A},
4075   a = {\^a,\^a,\^a,\^-a,\^a,\r a,\k a,\u a},
4076   C = {\^C,\^C,\v C},
4077   c = {\^c,\^c,\v c},
4078   D = {\v D,\DH},
4079   d = {\v d,\dj},
4080   E = {\^E,\^E,\^E,\^E,\^E,\k E,\v E},
4081   e = {\^e,\^e,\^e,\^e,\^e,\k e,\v e},
4082   f = {027}, % ff
4083   G = {\u G},
4084   g = {\u g},

```

```

4085 I = {\`I,\`I,\^I,\^I,\`I},
4086 i = {\`i,\`i,\^i,\^i,\`i},
4087 j = {\j},
4088 L = {\L,\`L,\v L},
4089 l = {\l,\`l,\v l},
4090 N = {\`N,\~N,\v N},
4091 n = {\`n,\~n,\v n},
4092 O = {\0,\`0,\`0,\^0,\~0,\`0,\H 0},
4093 o = {\0,\`o,\`o,\^o,\~o,\`o,\H o},
4094 R = {\`R,\v R},
4095 r = {\`r,\v r},
4096 S = {\`S,\c S,\v S,\SS},
4097 s = {\`s,\c s,\v s},
4098 T = {\c T,\v T},
4099 t = {\c t,\v t},
4100 U = {\`U,\`U,\^U,\^U,\H U,\r U},
4101 u = {\`u,\`u,\^u,\^u,\H u,\r u},
4102 Y = {\`Y,\`Y},
4103 y = {\`y,\`y},
4104 Z = {\`Z,\`Z,\v Z},
4105 z = {\`z,\`z,\v z}

```

The ‘soft hyphen’ often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```

4106 % - = {127},
4107 }
4108

```

15.5.3 LY1

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

```

4109 \DeclareCharacterInheritance
4110 { encoding = LY1 }
4111 { A = {\`A,\`A,\^A,\~A,\`A,\r A},
4112   a = {\`a,\`a,\^a,\~a,\`a,\r a},
4113   C = {\c C},
4114   c = {\c c},
4115   D = {\DH},
4116   E = {\`E,\`E,\^E,\`E},
4117   e = {\`e,\`e,\^e,\`e},
4118   f = {011}, % ff
4119   I = {\`I,\`I,\^I,\`I},
4120   i = {\`i,\`i,\^i,\`i},
4121   L = {\L},
4122   l = {\l},
4123   N = {\~N},
4124   n = {\~n},
4125   O = {\`O,\`O,\^O,\~O,\`O,\0},
4126   o = {\`o,\`o,\^o,\~o,\`o,\0},
4127   S = {\v S},
4128   s = {\v s},
4129   U = {\`U,\`U,\^U,\`U},
4130   u = {\`u,\`u,\^u,\`u},
4131   Y = {\`Y,\`Y},
4132   y = {\`y,\`y},
4133   Z = {\v Z},
4134   z = {\v z}
4135 }
4136

```

15.5.4 OT4

The Polish OT1 extension. More interesting characters here: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

4137 \DeclareCharacterInheritance
4138 { encoding = OT4 }
4139 { A = {\k A},
4140   a = {\k a},
4141   C = {\'C},
4142   c = {\'c},
4143   E = {\k E},
4144   e = {\k e},
4145   f = {011}, % ff
4146   i = {\i},
4147   j = {\j},
4148   L = {\L},
4149   l = {\l},
4150   N = {\'N},
4151   n = {\'n},
4152   O = {\0,\'0},
4153   o = {\o,\'o},
4154   S = {\'S},
4155   s = {\'s},
4156   Z = {\'Z,\.Z},
4157   z = {\'z,\.z}
4158 }
4159
```

15.5.5 QX

The Central European QX encoding.¹² Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

4160 \DeclareCharacterInheritance
4161 { encoding = QX }
4162 { A = {\^A,\'A,\^A,\-A,\"A,\k A,\AA},
4163   a = {\^a,\'a,\^a,\-a,\"a,\k a,\aa},
4164   C = {\'C,\c C},
4165   c = {\'c,\c c},
4166   D = {\DH},
4167   E = {\^E,\'E,\^E,\"E,\k E},
4168   e = {\^e,\'e,\^e,\"e,\k e},
4169   f = {011}, % ff
4170   I = {\^I,\'I,\^I,\"I,\k I},
4171   i = {\^i,\'i,\^i,\"i,\k i,\i},
4172   j = {\j},
4173   L = {\L},
4174   l = {\l},
4175   N = {\'N,\-N},
4176   n = {\'n,\-n},
4177   O = {\0,\'0,\^0,\-0,\"0},
4178   o = {\o,\'o,\^o,\-o,\"o},
4179   S = {\'S,\c S,\v S},
4180   s = {\'s,\c s,\v s},
4181   T = {\c T},
4182   t = {\c t},
4183   U = {\^U,\'U,\^U,\"U,\k U},
4184   u = {\^u,\'u,\^u,\"u,\k u},
4185   Y = {\'Y,\"Y},

```

¹² Thanks to *Maciej Eder*.

```

4186     y = {\'y,\"y},
4187     Z = {\'Z,\"Z,\"v Z},
4188     z = {\'z,\"z,\"v z},
4189     . = \textellipsis
4190 }
4191

```

15.5.6 T5

The Vietnamese encoding T5. It is so crowded with accented and double-accented characters that there is no room for any ligatures.

```

4192 \DeclareCharacterInheritance
4193 { encoding = T5 }
4194 { A = {\`A,\"A,\"-A,\"h A,\"d A,\"^A,\"u A,
4195       \`A\Acircumflex,\"A\Acircumflex,\"-A\Acircumflex,\"h\Acircumflex,\"d\Acircumflex,
4196       \`A\Abreve,\"A\Abreve,\"-A\Abreve,\"h\Abreve,\"d\Abreve},
4197   a = {\`a,\"a,\"-a,\"h a,\"d a,\"^a,\"u a,
4198       \`a\acircumflex,\"a\acircumflex,\"-a\acircumflex,\"h\acircumflex,\"d\acircumflex,
4199       \`a\abreve,\"a\abreve,\"-a\abreve,\"h\abreve,\"d\abreve},
4200   D = {\DJ},
4201   d = {\dj},
4202   E = {\`E,\"E,\"-E,\"h E,\"d E,\"^E,
4203       \`E\Ecircumflex,\"E\Ecircumflex,\"-E\Ecircumflex,\"h\Ecircumflex,\"d\Ecircumflex},
4204   e = {\`e,\"e,\"-e,\"h e,\"d e,\"^e,
4205       \`e\ecircumflex,\"e\ecircumflex,\"-e\ecircumflex,\"h\ecircumflex,\"d\ecircumflex},
4206   I = {\`I,\"I,\"-I,\"h I,\"d I},
4207   i = {\`i,\"i,\"-i,\"h i,\"d i,\"i},
4208   O = {\`O,\"O,\"-O,\"h O,\"d O,\"^O,\"horn O,
4209       \`O\Ocircumflex,\"O\Ocircumflex,\"-O\Ocircumflex,\"h\Ocircumflex,\"d\Ocircumflex,
4210       \`O\Ohorn,\"O\Ohorn,\"-O\Ohorn,\"h\Ohorn,\"d\Ohorn},
4211   o = {\`o,\"o,\"-o,\"h o,\"d o,\"^o,\"horn o,
4212       \`o\ocircumflex,\"o\ocircumflex,\"-o\ocircumflex,\"h\ocircumflex,\"d\ocircumflex,
4213       \`o\ohorn,\"o\ohorn,\"-o\ohorn,\"h\ohorn,\"d\ohorn},
4214   U = {\`U,\"U,\"-U,\"h U,\"d U,\"horn U,
4215       \`U\Uhorn,\"U\Uhorn,\"-U\Uhorn,\"h\Uhorn,\"d\Uhorn},
4216   u = {\`u,\"u,\"-u,\"h u,\"d u,\"horn u,
4217       \`u\uhorn,\"u\uhorn,\"-u\uhorn,\"h\uhorn,\"d\uhorn},
4218   Y = {\`Y,\"Y,\"-Y,\"h Y,\"d Y},
4219   y = {\`y,\"y,\"-y,\"h y,\"d y}
4220 }
4221
4222 /m-t

```

15.5.7 Euro symbols

Make Euro symbols settings simpler.

```

4223 *zpeu
4224 \DeclareCharacterInheritance
4225 { encoding = U,
4226   family   = {zpeu,zpeus,eurosans} }
4227 { E = 128 }
4228
4229 /zpeu
4230 *mvs
4231 \DeclareCharacterInheritance
4232 { encoding = OT1,
4233   family   = mvs }
4234 { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
4235

```

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

```
4236 \DeclareCharacterInheritance
4237   { encoding = U,
4238     family   = mvs }
4239   { 164 = {099,100,101} }
4240
4241 (mvs)
```

15.6 Tracking

By default, we only disable the ‘f*’ ligatures, for those fonts that have any. Thus, ligatures and especially kerning for all other characters will be retained.

```
4242 (*m-t)
4243 %%% -----
4244 %%% TRACKING/LETTERSPACING
4245
4246 \SetTracking
4247   [ name      = default,
4248     no ligatures = {f} ]
4249   { encoding   = {OT1,T1,LY1,OT4,QX} }
4250   { }
4251
```

15.7 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).

```
4252 %%% -----
4253 %%% EXPANSION
4254
4255 \SetExpansion
4256   [ name      = default      ]
4257   { encoding   = {OT1,OT4,QX,T1,LY1} }
4258   {
4259     A = 500,      a = 700,
4260     \AE = 500,    \ae = 700,
4261     B = 700,      b = 700,
4262     C = 700,      c = 700,
4263     D = 500,      d = 700,
4264     E = 700,      e = 700,
4265     F = 700,
4266     G = 500,      g = 700,
4267     H = 700,      h = 700,
4268     K = 700,      k = 700,
4269     M = 700,      m = 700,
4270     N = 700,      n = 700,
4271     O = 500,      o = 700,
4272     \OE = 500,    \oe = 700,
4273     P = 700,      p = 700,
4274     Q = 500,      q = 700,
4275     R = 700,
4276     S = 700,      s = 700,
4277     U = 700,      u = 700,
4278     W = 700,      w = 700,
4279     Z = 700,      z = 700,
4280     2 = 700,
```

```

4281     3 = 700,
4282     6 = 700,
4283     8 = 700,
4284     9 = 700
4285   }
4286

```

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

```

4287 \SetExpansion
4288   [ name      = T5 ]
4289   { encoding = T5 }
4290   {
4291     A = 500,      a = 700,
4292     B = 700,      b = 700,
4293     C = 700,      c = 700,
4294     D = 500,      d = 700,
4295     E = 700,      e = 700,
4296     F = 700,
4297     G = 500,      g = 700,
4298     H = 700,      h = 700,
4299     K = 700,      k = 700,
4300     M = 700,      m = 700,
4301     N = 700,      n = 700,
4302     O = 500,      o = 700,
4303     P = 700,      p = 700,
4304     Q = 500,      q = 700,
4305     R = 700,
4306     S = 700,      s = 700,
4307     U = 700,      u = 700,
4308     W = 700,      w = 700,
4309     Z = 700,      z = 700,
4310     2 = 700,
4311     3 = 700,
4312     6 = 700,
4313     8 = 700,
4314     9 = 700
4315   }
4316
4317 </m-t>

```

15.8 Character protrusion

```

4318 %%% -----
4319 %%% PROTRUSION
4320

```

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

```

\SetProtrusion
[ name      = thanh ]
{ encoding = OT1 }
{
  A = {50,50},
  F = { ,50},
  J = {50, },
  K = { ,50},
  L = { ,50},
  T = {50,50},
  V = {50,50},
  W = {50,50},
  X = {50,50},

```

```

Y = {50,50},
k = { ,50},
r = { ,50},
t = { ,50},
v = {50,50},
w = {50,50},
x = {50,50},
y = {50,50},
. = { ,700},    {,}= { ,700},
: = { ,500},    ; = { ,500},
! = { ,200},    ? = { ,200},
( = {50, },    ) = { ,50},
- = { ,700},
\textendash     = { ,300},    \textendash     = { ,200},
\textquoteleft  = {700, },    \textquoteright = { ,700},
\textquotedblleft = {500, },    \textquotedblright = { ,500}
}

```

15.8.1 Default

The default settings always use the most moderate value.

```

4321 <*cfg-t>
4322 \SetProtrusion
4323 <m-t> [ name = default ]

```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```

4324 <bch> [ name = bch-default ]

```

- Bitstream Letter Gothic (blg)

```

4325 <blg> [ name = blg-default ]

```

- Computer Modern Roman (cmr)

```

4326 <cmr> [ name = cmr-default ]

```

- Adobe Garamond (pad, padx, padj)

```

4327 <pad> [ name = pad-default ]

```

- Minion¹³ (pmnx, pmnj)

```

4328 <pmn> [ name = pmnj-default ]

```

- Palatino (ppl, pplx, pplj)

```

4329 <ppl> [ name = ppl-default ]

```

- Times (ptm, ptmx, ptmj)

```

4330 <ptm> [ name = ptm-default ]

```

- URW Garamond (ugm)

```

4331 <ugm> [ name = ugm-default ]
4332 <m-t> { encoding = OT1 }
4333 <cmr> { }
4334 <bch|blg|pad|pmn|ugm> { encoding = OT1,
4335 <ppl|ptm> { encoding = {OT1,OT4},

```

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

```

4336 <bch>      family   = bch }
4337 <blg>      family   = blg }
4338 <pad>      family   = {pad,padx,padj} }
4339 <pmn>      family   = pmnj }
4340 <ppl>      family   = {ppl,pplx,pplj} }
4341 <ptm>      family   = {ptm,ptmx,ptmj} }
4342 <ugm>      family   = ugm }
4343 {
4344 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm>    A = {50,50},
4345 <ugm>      A = {50,100},
4346 <m-t|pad|ptm>    \AE = {50, },
4347 <ugm>      \AE = {150,50},
4348 <ugm>      B = { ,50},
4349 <bch|pad|pmn|ugm>    C = {50, },
4350 <bch|pad|pmn>      D = { ,50},
4351 <ugm>      D = { ,70},
4352 <ugm>      E = { ,50},
4353 <m-t|bch|cmr|pad|pmn|ptm>    F = { ,50},
4354 <ugm>      F = { ,70},
4355 <bch|pad|pmn>      G = {50, },
4356 <ugm>      G = {50,50},
4357 <blg>      I = {150,150},
4358 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    J = {50, },
4359 <bch|blg>      J = {100, },
4360 <!blg>      K = { ,50},
4361 <blg>      K = {50, },
4362 <m-t|bch|cmr|pad|pmn|ppl>    L = { ,50},
4363 <blg>      L = { ,150},
4364 <ptm>      L = { ,80},
4365 <ugm>      L = { ,120},
4366 <bch|pad|pmn|ugm>    O = {50,50},
4367 <pad|pmn>      \OE = {50, },
4368 <ugm>      \OE = {50,50},
4369 <blg>      P = { ,100},
4370 <ugm>      P = { ,50},
4371 <bch|pad|pmn>      Q = {50,70},
4372 <ugm>      Q = {50,50},
4373 <bch>      R = { ,50},
4374 <ugm>      R = { ,70},
4375 <m-t|bch|cmr|pad|pmn|ppl|ptm>    T = {50,50},
4376 <blg>      T = {100,100},
4377 <ugm>      T = {70,70},
4378 <m-t|bch|cmr|pad|pmn|ppl|ptm>    V = {50,50},
4379 <blg|ugm>      V = {70,70},
4380 <m-t|bch|cmr|pad|pmn|ppl|ptm>    W = {50,50},
4381 <ugm>      W = {70,70},
4382 <m-t|bch|cmr|pad|pmn|ppl|ptm>    X = {50,50},
4383 <ugm>      X = {50,70},
4384 <m-t|bch|cmr|pad|pmn|ppl>      Y = {50,50},
4385 <blg|ptm|ugm>      Y = {80,80},
4386 <ugm>      Z = {50,50},
4387 <blg>      f = {150,100},
4388 <blg>      i = {150,150},
4389 <blg>      j = {100,100},
4390 <m-t|bch|cmr|pad|pmn|ppl|ptm>    k = { ,50},
4391 <ugm>      k = { ,70},
4392 <blg>      l = {150,150},
4393 <pmn>      l = { , -50},
4394 <pad|ppl>      p = {50,50},
4395 <ugm>      p = { ,50},
4396 <pad|ppl>      q = {50, },
4397 <!blg>      r = { ,50},
4398 <blg>      r = {100, 80},

```



```

4399 <cmr|pad|pmn>      t = { ,70},
4400 <bch>               t = { ,50},
4401 <blg>               t = {150, 80},
4402 <ugm>               t = { ,100},
4403 <m-t|bch|cmr|pad|pmn|ppl|ptm>    v = {50,50},
4404 <blg>               v = {100,100},
4405 <ugm>               v = {50,70},
4406 <m-t|bch|cmr|pad|pmn|ppl|ptm>    w = {50,50},
4407 <ugm>               w = {50,70},
4408 <!blg>              x = {50,50},
4409 <blg>               x = {100,100},
4410 <m-t|bch|pad|pmn>   y = { ,50},
4411 <blg>               y = { 50,100},
4412 <cmr|ppl|ptm>      y = {50,70},
4413 <ugm>               y = { ,70},

4414 <cmr>               0 = { ,50},
4415 <m-t>               1 = {50,50},
4416 <bch|blg|pad|ptm|ugm>      1 = {150,150},
4417 <cmr>               1 = {100,200},
4418 <pmn>               1 = { ,50},
4419 <ppl>                1 = {100,100},
4420 <bch|cmr|pad|ugm>     2 = {50,50},
4421 <blg>               2 = { ,100},
4422 <bch|pmn>           3 = {50, },
4423 <cmr|pad|ugm>        3 = {50,50},
4424 <blg>               3 = {100, },
4425 <m-t|pad>           4 = {50,50},
4426 <bch>               4 = {100,50},
4427 <blg>               4 = {100, },
4428 <cmr|ugm>           4 = {70,70},
4429 <pmn>               4 = {50, },
4430 <ptm>               4 = {70, },
4431 <cmr>               5 = { ,50},
4432 <pad>               5 = {50,50},
4433 <bch>               6 = {50, },
4434 <cmr>               6 = { ,50},
4435 <pad>               6 = {50,50},
4436 <m-t>               7 = {50,50},
4437 <bch|pad|pmn|ugm>     7 = {50,80},
4438 <blg>               7 = {100,100},
4439 <cmr|ptm>           7 = {50,100},
4440 <ppl>               7 = { ,50},
4441 <cmr>               8 = { ,50},
4442 <bch|pad>           9 = {50,50},
4443 <cmr>               9 = { ,50},
4444 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    . = { ,700},
4445 <bch>               . = { ,600},
4446 <blg>               . = {400,500},
4447 <!blg>              {,}= { ,500},
4448 <blg>               {,}= {300,400},
4449 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    : = { ,500},
4450 <bch>               : = { ,400},
4451 <blg>               : = {300,400},
4452 <m-t|bch|pad|pmn|ptm>    ; = { ,300},
4453 <blg>               ; = {200,300},
4454 <cmr|ppl>           ; = { ,500},
4455 <ugm>               ; = { ,400},
4456 <!blg>              != { ,100},
4457 <blg>               != {200,200},
4458 <m-t|pad|pmn|ptm>     ? = { ,100},
4459 <bch|cmr|ppl|ugm>     ? = { ,200},
4460 <blg>               ? = {150,150},

```

```

4461 <pmn> " = {300,300},
4462 <m-t|bch|cmr|pad|pmn|ppl> @ = {50,50},
4463 <ptm> @ = {100,100},
4464 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> ~ = {200,250},
4465 <ugm> ~ = {300,350},
4466 <pad|ppl|ptm> & = {50,100},
4467 <ugm> & = { ,100},
4468 <m-t|cmr|pad|pmn> \% = {50,50},
4469 <bch> \% = { ,50},
4470 <ppl|ptm> \% = {100,100},
4471 <ugm> \% = {50,100},
4472 <blg> \# = {100,100},
4473 <m-t|ppl|ptm|ugm> * = {200,200},
4474 <bch|pmn> * = {200,300},
4475 <blg> * = {150,200},
4476 <cmr|pad> * = {300,300},
4477 <m-t|cmr|ppl|ptm> + = {250,250},
4478 <bch> + = {150,250},
4479 <pad> + = {300,300},
4480 <blg|pmn> + = {150,200},
4481 <ugm> + = {250,300},
4482 <blg|ugm> {=} = {200,200},
4483 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
4484 <bch|ugm> ( = {200, }, ) = { ,200},
4485 <cmr|blg> ( = {300, }, ) = { ,300},
4486 <ppl> ( = {100, }, ) = { ,300},
4487 <bch|pmn> [ = {100, }, ] = { ,100},
4488 <blg> [ = {300,100}, ] = { ,300},

4489 <m-t|pad|pmn|ptm> / = {100,200},
4490 <bch> / = { ,200},
4491 <blg> / = {300,300},
4492 <cmr|ppl> / = {200,300},
4493 <ugm> / = {100,300},
4494 <m-t|ptm> - = {500,500},
4495 <bch|cmr|ppl> - = {400,500},
4496 <blg> - = {300,400},
4497 <pad> - = {300,500},
4498 <pmn> - = {200,400},
4499 <ugm> - = {500,600},
4500 <blg> <= {200,100}, >= {100,200},
4501 <blg> - = {150,250},
4502 <blg> | = {250,250},
4503 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
4504 <bch> \textendash = {200,300}, \textendash = {150,250},
4505 <cmr> \textendash = {400,300}, \textendash = {300,200},
4506 <pad|ppl|ptm> \textendash = {300,300}, \textendash = {200,200},
4507 <ugm> \textendash = {250,300}, \textendash = {250,250},

```

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

```

4508 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
4509 <blg> \textquoteleft = {400,600}, \textquoteright = {400,600},
4510 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
4511 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
4512 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
4513 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
4514 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300}
4515 <blg> \textquotedblright = {300,400}
4516 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600}
4517 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4518 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400}

```

```
4519 }
4520
```

Greek uppercase letters are in OT1 encoding only.

```
4521 (*cmr)
4522 \SetProtrusion
4523 [ name      = cmr-OT1,
4524   load      = cmr-default ]
4525 { encoding = {OT1,OT4},
4526   family    = cmr }
4527 {
4528   \AE = { 50, },
4529   "00 = { ,150}, % \Gamma
4530   "01 = {100,100}, % \Delta
4531   "02 = { 50, 50}, % \Theta
4532   "03 = {100,100}, % \Lambda
4533   "06 = { 50, 50}, % \Sigma
4534   "07 = {100,100}, % \Upsilon
4535   "08 = { 50, 50}, % \Phi
4536   "09 = { 50, 50} % \Psi
```

Remaining slots can be found in the source file.

```
4537 }
4538
4539 </cmr>
```

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

```
4540 \SetProtrusion
4541 <m-t> [ name      = T1-default,
4542 <bch> [ name      = bch-T1,
4543 <blg> [ name      = blg-T1,
4544 <cmr> [ name      = cmr-T1,
4545 <pad> [ name      = pad-T1,
4546 <pmn> [ name      = pmnj-T1,
4547 <ppl> [ name      = ppl-T1,
4548 <ptm> [ name      = ptm-T1,
4549 <ugm> [ name      = ugm-T1,
4550 <m-t>   load      = default ]
4551 <bch>   load      = bch-default ]
4552 <blg>   load      = blg-default ]
4553 <cmr>   load      = cmr-default ]
4554 <pad>   load      = pad-default ]
4555 <pmn>   load      = pmnj-default ]
4556 <ppl>   load      = ppl-default ]
4557 <ptm>   load      = ptm-default ]
4558 <ugm>   load      = ugm-default ]
4559 <m-t>   { encoding = {T1,LY1} }
4560 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4561 <blg|ptm|ugm> { encoding = {T1},
4562 <bch>   family    = bch }
4563 <blg>   family    = blg }
4564 <cmr>   family    = cmr }
4565 <pad>   family    = {pad,padx,padj} }
4566 <pmn>   family    = pmnj }
4567 <ppl>   family    = {ppl,pplx,pplj} }
4568 <ptm>   family    = {ptm,ptmx,ptmj} }
4569 <ugm>   family    = ugm }
4570 {
4571 <cmr>   \AE = {50, },
4572 <bch>   \OE = {50, },
4573 <pmn>   \TH = { ,50},
```

```

4574 <blg> \v L = { ,250},
4575 <blg> \v d = { ,250},
4576 <blg> \v l = { ,250},
4577 <blg> \v t = { ,250},
4578 <blg> 127 = {300,400},
4579 <blg> 156 = {100, }, % IJ
4580 <blg> 188 = { 80, 80}, % ij
4581 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
4582 <cmr> _ = {200,200},
4583 <ugm> _ = {100,200},
4584 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
4585 <bch> \textbackslash = {150,200},
4586 <blg> \textbackslash = {250,300},
4587 <cmr|ppl> \textbackslash = {200,300},
4588 <ugm> \textbackslash = {100,300},
4589 <ugm> \textbar = {200,200},
4590 <blg> \textendash = {300,300}, \textemdash = {150,150},
4591 <blg> \textquotedbl = {300,400}, \textquotedblleft = {300,400},
4592 <cmr> \textquotedbl = {300,300}, \textquotedblleft = {200,600},

```

The EC fonts do something weird: they insert an implicit kern between quote and boundary character. Therefore, we must override the settings from OT1.

```

4593 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4594 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
4595 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4596 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsingright = {300,400},
4597 <blg> \guilsinglleft = {300,500}, \guilsingright = {300,500},
4598 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsingright = {300,500},
4599 <ugm> \guilsinglleft = {400,400}, \guilsingright = {300,600},
4600 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
4601 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
4602 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
4603 <blg|pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
4604 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4605 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
4606 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
4607 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4608 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
4609 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
4610 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
4611 <pmn> \textless = {100, }, \textgreater = { ,100},
4612 <pmn> \textvisiblespace = {100,100} % not in LY1
4613 }
4614

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```

4615 <*cmr>
4616 \SetProtrusion
4617 [ name = lmr-T1,
4618   load = cmr-T1 ]
4619 { encoding = {T1,LY1},
4620   family = lmr }
4621 {
4622   \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4623 }
4624
4625 </cmr>

```

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

```

4626 <*m-t|ptm>
4627 \SetProtrusion
4628 <m-t> [ name = QX-default,
4629 <ptm> [ name = ptm-QX,
4630 <m-t> load = default ]
4631 <ptm> load = ptm-default ]
4632 <m-t> { encoding = QX }
4633 <ptm> { encoding = QX,
4634 <ptm> family = {ptm,ptmx,ptmj} }
4635 {
4636 <ptm> * = {200,200},
4637 {=} = {100,100},
4638 \textunderscore = {100,100},
4639 \textbackslash = {100,200},
4640 \quotedblbase = {400,400},
4641 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
4642 <ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
4643 \textexclamdown = {100, }, \textquestiondown = {100, },
4644 <m-t> \textbraceleft = {400,200}, \textbraceright = {200,400},
4645 <ptm> \textbraceleft = {200,200}, \textbraceright = {200,300},
4646 \textless = {200,100}, \textgreater = {100,200},
4647 \textminus = {200,200}, \textdegree = {300,300},
4648 <m-t> \copyright = {100,100}, \textregistered = {100,100}
4649 <ptm> \copyright = {100,150}, \textregistered = {100,150},
4650 <ptm> \textxgeq = { ,100}, \textxleq = {100, },
4651 <ptm> \textalpha = { , 50}, \textDelta = { 70, 70},
4652 <ptm> \textpi = { 50, 80}, \textSigma = { , 70},
4653 <ptm> \textmu = { , 80}, \texteuro = { 50, 50},
4654 <ptm> \textellipsis = {150,200}, \textasciitilde = { 80, 80},
4655 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
4656 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
4657 <ptm> \textdiv = { 50,150}, \textsection = { 80, 80},
4658 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
4659 <ptm> \textbullet = {150,150}, \textperiodcentered = {300,300},
4660 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
4661 <ptm> \textperthousand = { ,50}
4662 }
4663
4664 </m-t|ptm>

```

T5 is based on OT1; it shares some but not all extra characters of T1. All accented characters are already taken care of by the inheritance list.

```

4665 <*cmr|bch>
4666 \SetProtrusion
4667 <cmr> [ name = cmr-T5,
4668 <cmr> load = cmr-default ]
4669 <bch> [ name = bch-T5,
4670 <bch> load = bch-default ]
4671 { encoding = T5,
4672 <cmr> family = cmr }
4673 <bch> family = bch }
4674 {
4675 <bch> _ = {100,100},
4676 <bch> \textbackslash = {150,200},
4677 <cmr> \textbackslash = {200,300},
4678 <cmr> \textquotedblleft = {200,600},
4679 <cmr> \textquotedbl = {300,300},
4680 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4681 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4682 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4683 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4684 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},

```

```

4685 <cmr>    \guillemotleft  = {300,200}, \guillemotright = {100,400},
4686 <bch>    \textbraceleft   = {200,  }, \textbraceright  = {  ,300},
4687 <cmr>    \textbraceleft   = {400,200}, \textbraceright  = {200,400},
4688 \textless    = {200,100}, \textgreater    = {100,200}
4689 }
4690
4691 </cmr|bch>
4692 <*pmn>
4693 \SetProtrusion
4694 [ name      = pmnx-OT1,
4695   load      = pmnj-default ]
4696 { encoding = OT1,
4697   family   = pmnx }
4698 {
4699   1 = {230,180}
4700 }
4701
4702 \SetProtrusion
4703 [ name      = pmnx-T1,
4704   load      = pmnj-T1 ]
4705 { encoding = {T1,LY1},
4706   family   = pmnx }
4707 {
4708   1 = {230,180}
4709 }
4710
4711 </pmn>

```

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

```

4712 <*ptm>
4713 \SetProtrusion
4714 [ name      = ptm-LY1,
4715   load      = ptm-T1 ]
4716 { encoding = LY1,
4717   family   = {ptm,ptmx,ptmj} }
4718 {
4719   - = {100,100},
4720 \texttrademark = {100,100},
4721 \textregistered = {100,100},
4722 \textcopyright = {100,100},
4723 \textdegree = {300,300},
4724 \textminus = {200,200},
4725 \textellipsis = {150,200},
4726 \texteuro = {  ,  }, % ?
4727 \textcent = {100,100},
4728 \textquotesingle = {500,500},
4729 \textflorin = { 50, 70},
4730 \textdagger = {150,150},
4731 \textdaggerdbl = {100,100},
4732 \textperthousand = {  , 50},
4733 \textbullet = {150,150},
4734 \textonesuperior = {100,100},
4735 \texttwosuperior = { 50, 50},
4736 \textthreesuperior = { 50, 50},
4737 \textperiodcentered = {300,300},
4738 \textplusminus = { 50, 80},
4739 \textmultiply = {100,100},
4740 \textdivide = { 50,150}

```

Remaining slots in the source file.

```

4741 }

```

```
4742
4743 </ptm>
```

15.8.2 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.

```
4744 \SetProtrusion
4745 <m-t> [ name      = OT1-it   ]
4746 <bch> [ name      = bch-it   ]
4747 <blg> [ name      = blg-it,
4748 <blg>   load      = blg-default ]
4749 <cmr> [ name      = cmr-it   ]
4750 <pad> [ name      = pad-it   ]
4751 <pmn> [ name      = pmnj-it  ]
4752 <ppl> [ name      = ppl-it   ]
4753 <ptm> [ name      = ptm-it   ]
4754 <ugm> [ name      = ugm-it   ]
4755 <m-t|bch|blg|pad|pmn|ugm> { encoding = OT1,
4756 <ppl|ptm> { encoding = {OT1,OT4},
4757 <bch>      family   = bch,
4758 <blg>      family   = blg,
4759 <pad>      family   = {pad,padx,padj},
4760 <pmn>      family   = pmnj,
4761 <ppl>      family   = {ppl,pplx,pplj},
4762 <ptm>      family   = {ptm,ptmx,ptmj},
4763 <ugm>      family   = ugm,
4764 <m-t|bch|pad|pmn|ppl|ptm> shape   = {it,sl} }
4765 <blg|ugm> shape   = it }
4766 <cmr> { }
4767 {
4768 <cmr|ptm>   A = {100,50},
4769 <pad|pmn>   A = {50, },
4770 <ugm>       A = { ,150},
4771 <ppl>       A = {50,50},
4772 <ptm>       \AE = {100, },
4773 <pad|ppl>   \AE = {50, },
4774 <pmn>       \AE = { , -50},
4775 <cmr|pad|ppl|ptm> B = {50, },
4776 <pmn>       B = {20,-50},
4777 <bch|ppl|ptm|ugm> C = {50, },
4778 <cmr|pad>   C = {100, },
4779 <pmn>       C = {50,-50},
4780 <cmr|pad|ppl|ptm> D = {50,50},
4781 <pmn>       D = {20, },
4782 <cmr|pad|ppl|ptm> E = {50, },
4783 <pmn>       E = {20,-50},
4784 <cmr|pad|ptm> F = {100, },
4785 <pmn>       F = {10, },
4786 <ppl>       F = {50, },
4787 <bch|ppl|ptm|ugm> G = {50, },
4788 <cmr|pad>   G = {100, },
4789 <pmn>       G = {50,-50},
4790 <cmr|pad|ppl|ptm> H = {50, },
4791 <cmr|pad|ptm> I = {50, },
4792 <pmn>       I = {20,-50},
4793 <cmr|ptm>   J = {100, },
4794 <pad>       J = {50, },
4795 <pmn>       J = {20, },
4796 <cmr|pad|ppl|ptm> K = {50, },
```

```

4797 <pmn>      K = {20,  },
4798 <cmr|pad|ppl|ptm>  L = {50,  },
4799 <pmn>      L = {20,50},
4800 <ugm>      L = {  ,100},
4801 <cmr|ptm>   M = {50,  },
4802 <pmn>      M = {  ,-30},
4803 <cmr|ptm>   N = {50,  },
4804 <pmn>      N = {  ,-30},
4805 <bch|pmn|ppl|ptm>  O = {50,  },
4806 <cmr|pad>   O = {100,  },
4807 <ugm>      O = {70,50},
4808 <pmn|ppl|ptm> \OE = {50,  },
4809 <pad>       \OE = {100,  },
4810 <cmr|pad|ppl|ptm>  P = {50,  },
4811 <pmn>      P = {20,-50},
4812 <bch|pmn|ppl|ptm>  Q = {50,  },
4813 <cmr|pad>   Q = {100,  },
4814 <ugm>      Q = {70,50},
4815 <cmr|pad|ppl|ptm>  R = {50,  },
4816 <pmn>      R = {20,  },
4817 <bch|cmr|pad|ppl|ptm> S = {50,  },
4818 <pmn>      S = {20,-30},
4819 <bch|cmr|pad|ppl|ptm> $ = {50,  },
4820 <pmn>      $ = {20,-30},
4821 <bch|pmn|ugm>      T = {70,  },
4822 <cmr|pad|ppl|ptm>  T = {100,  },
4823 <cmr|pad|ppl|ptm>  U = {50,  },
4824 <pmn>      U = {50,-50},
4825 <cmr|pad|pmn|ugm>  V = {100,  },
4826 <ppl|ptm>      V = {100,50},
4827 <cmr|pad|pmn|ugm>  W = {100,  },
4828 <ppl>         W = {50,  },
4829 <ptm>         W = {100,50},
4830 <cmr|ppl|ptm>     X = {50,  },
4831 <cmr|ptm>       Y = {100,  },
4832 <pmn>         Y = {50,  },
4833 <ppl>         Y = {100,50},
4834 <pmn>         Z = {  ,-50},
4835 <pmn>         d = {  ,-50},
4836 <pad|pmn>      f = {  ,-100},
4837 <pmn>         i = {  ,-30},
4838 <pmn>         j = {  ,-30},
4839 <pmn>         l = {  ,-100},
4840 <bch>         o = {50,50},
4841 <bch>         p = {  ,50},
4842 <pmn>         p = {-50,  },
4843 <bch>         q = {50,  },
4844 <pmn>         r = {  ,50},
4845 <bch>         t = {  ,50},
4846 <pmn|ugm>     v = {50,  },
4847 <bch>         w = {  ,50},
4848 <pmn|ugm>     w = {50,  },
4849 <bch>         y = {  ,50},
4850 <cmr>         0 = {100,  },
4851 <bch|ptm>      1 = {150,100},
4852 <cmr>         1 = {200,50},
4853 <pad>         1 = {150,  },
4854 <pmn>         1 = {50,  },
4855 <ppl>         1 = {100,  },
4856 <ugm>         1 = {150,150},
4857 <cmr>         2 = {100,-100},
4858 <pad|ppl|ptm>    2 = {50,  },
4859 <pmn>         2 = {-50,  },

```



```

4860 <bch>      3 = {50, },
4861 <cmr>      3 = {100,-100},
4862 <pmn>      3 = {-100, },
4863 <ptm>      3 = {100,50},
4864 <bch>      4 = {100, },
4865 <cmr|pad>  4 = {150, },
4866 <ppl|ptm>  4 = {50, },
4867 <cmr>      5 = {100, },
4868 <ptm>      5 = {50, },
4869 <bch>      6 = {50, },
4870 <cmr>      6 = {100, },
4871 <bch|pad|ptm> 7 = {100, },
4872 <cmr>      7 = {200,-150},
4873 <pmn>      7 = {20, },
4874 <ppl>      7 = {50, },
4875 <cmr>      8 = {50,-50},
4876 <cmr>      9 = {100,-100},
4877 <m-t|cmr|pad|pmn|ppl> . = { ,500},
4878 <blg>      . = {400,600},
4879 <bch|ptm|ugm> . = { ,700},
4880 <blg>      {,}= {300,500},
4881 <m-t|cmr|pad|pmn|ppl> {,}= { ,500},
4882 <bch|ugm>   {,}= { ,600},
4883 <ptm>      {,}= { ,700},
4884 <m-t|cmr|pad|ppl>   : = { ,300},
4885 <bch|ugm>   : = { ,400},
4886 <pmn>      : = { ,200},
4887 <ptm>      : = { ,500},
4888 <m-t|cmr|pad|ppl>   ; = { ,300},
4889 <bch|ugm>   ; = { ,400},
4890 <pmn>      ; = { ,200},
4891 <ptm>      ; = { ,500},
4892 <ptm>      ! = { ,100},
4893 <bch>      ? = { ,200},
4894 <ptm>      ? = { ,100},
4895 <ppl>      ? = { ,300},
4896 <pmn>      " = {400,200},
4897 <m-t|pad|pmn|ppl|ptm> & = {50,50},
4898 <bch>      & = { ,80},
4899 <cmr>      & = {100,50},
4900 <ugm>      & = {50,100},
4901 <m-t|cmr|pad|pmn>   \% = {100, },
4902 <bch>      \% = {50,50},
4903 <ppl|ptm>   \% = {100,100},
4904 <ugm>      \% = {100,50},
4905 <m-t|pmn|ppl>      * = {200,200},
4906 <bch>      * = {300,200},
4907 <cmr>      * = {400,100},
4908 <pad>      * = {500,100},
4909 <ptm|ugm>   * = {400,200},
4910 <m-t|cmr|pmn|ppl>   + = {150,200},
4911 <bch|ugm>   + = {250,250},
4912 <pad|ptm>   + = {250,200},
4913 <m-t|pad|pmn|ppl>   @ = {50,50},
4914 <bch>      @ = {80,50},
4915 <cmr>      @ = {200,50},
4916 <ptm>      @ = {150,150},
4917 <m-t|bch|ugm>      ~ = {150,150},
4918 <cmr|pad|pmn|ppl|ptm> ~ = {200,150},
4919 <ugm>      {=}= {200,200},
4920 <!blg>      ( = {200, }, ) = { ,200},
4921 <m-t|cmr|pad|ppl|ptm|ugm> / = {100,200},
4922 <bch>      / = { ,150},

```

```

4923 <pmn>      / = {100,150},
4924 <m-t>      - = {300,300},
4925 <bch|pad>   - = {300,400},
4926 <pmn>      - = {200,300},
4927 <cmr>      - = {500,300},
4928 <ppl>      - = {300,500},
4929 <ptm>      - = {500,500},
4930 <ugm>      - = {400,700},
4931 <blg>      - = {0,300},
4932 <m-t|pmn>   \textendash      = {200,200}, \textendash      = {150,150},
4933 <bch>       \textendash      = {200,300}, \textendash      = {150,200},
4934 <cmr>       \textendash      = {500,300}, \textendash      = {400,200},
4935 <pad|ppl|ptm|ugm> \textendash      = {300,300}, \textendash      = {200,200},
4936 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
4937 <blg>       \textquoteleft = {400,400}, \textquoteright = {400,400},
4938 <cmr|pad>   \textquoteleft = {800,200}, \textquoteright = {800,200},
4939 <ppl>       \textquoteleft = {700,400}, \textquoteright = {700,400},
4940 <ptm>       \textquoteleft = {800,500}, \textquoteright = {800,500},
4941 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
4942 <blg>       \textquotedblright = {300,300}
4943 <cmr>       \textquotedblleft = {700,100}, \textquotedblright = {500,300}
4944 <pad>       \textquotedblleft = {700,200}, \textquotedblright = {700,200}
4945 <ppl>       \textquotedblleft = {500,300}, \textquotedblright = {500,300}
4946 <ptm>       \textquotedblleft = {700,400}, \textquotedblright = {700,400}
4947 <ugm>       \textquotedblleft = {600,200}, \textquotedblright = {600,200}
4948   }
4949
4950 <*cmr>
4951 \SetProtrusion
4952   [ name      = cmr-it-OT1,
4953     load      = cmr-it  ]
4954   { encoding = {OT1,OT4},
4955     family   = cmr,
4956     shape    = it      }
4957   {
4958     \AE = {100,  },
4959     \OE = {100,  },
4960     "00 = {200,150}, % \Gamma
4961     "01 = {150,100}, % \Delta
4962     "02 = {150, 50}, % \Theta
4963     "03 = {150, 50}, % \Lambda
4964     "04 = {100,100}, % \Xi
4965     "05 = {100,100}, % \Pi
4966     "06 = {100, 50}, % \Sigma
4967     "07 = {200,150}, % \Upsilon
4968     "08 = {150, 50}, % \Phi
4969     "09 = {150,100}, % \Psi
4970     "0A = { 50, 50} % \Omega
4971   }
4972
4973 </cmr>
4974 \SetProtrusion
4975 <m-t> [ name      = T1-it-default,
4976 <bch> [ name      = bch-it-T1,
4977 <blg> [ name      = blg-it-T1,
4978 <cmr> [ name      = cmr-it-T1,
4979 <pad> [ name      = pad-it-T1,
4980 <pmn> [ name      = pmn-it-T1,
4981 <ppl> [ name      = ppl-it-T1,
4982 <ptm> [ name      = ptm-it-T1,
4983 <ugm> [ name      = ugm-it-T1,
4984 <m-t> [ load      = OT1-it  ]
4985 <bch> [ load      = bch-it  ]

```

```

4986 <blg>      load      = blg-T1   ]
4987 <cmr>      load      = cmr-it    ]
4988 <pmn>      load      = pmnj-it   ]
4989 <pad>      load      = pad-it    ]
4990 <ppl>      load      = ppl-it     ]
4991 <ptm>      load      = ptm-it    ]
4992 <ugm>      load      = ugm-it    ]
4993 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4994 <blg|ptm|ugm> { encoding = T1,
4995 <bch>      family    = bch,
4996 <blg>      family    = blg,
4997 <cmr>      family    = cmr,
4998 <pmn>      family    = pmnj,
4999 <pad>      family    = {pad,padx,padj},
5000 <ppl>      family    = {ppl,pplx,pplj},
5001 <ptm>      family    = {ptm,ptmx,ptmj},
5002 <ugm>      family    = ugm,
5003 <m-t|bch|pad|pmn|ppl|ptm> shape    = {it,sl}  }
5004 <blg|cmr|ugm> shape    = it      }
5005 {
5006 <m-t|bch|pmn>      _ = { ,100},
5007 <blg>              _ = {0,300},
5008 <cmr|ugm>          _ = {100,200},
5009 <pad|ppl|ptm>      _ = {100,100},
5010 <blg>              . = {400,600},
5011 <blg>              {,}= {300,500},
5012 <cmr>              \AE = {100,  },
5013 <bch>              \OE = { 50,  },
5014 <cmr>              \OE = {100,  },
5015 <pmn>              031 = { , -100}, % ffl
5016 <cmr|ptm>          156 = {100, }, % IJ
5017 <pad>              156 = {50,  }, % IJ
5018 <pmn>              156 = {20,  }, % IJ
5019 <pmn>              188 = { , -30}, % ij
5020 <pmn>              \v t = { ,100},
5021 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
5022 <cmr|ugm>          \textbackslash = {300,300},
5023 <bch>              \textbackslash = {150,150},
5024 <pmn>              \textbackslash = {100,150},
5025 <ugm>              \textbar      = {200,200},
5026 <cmr>              \textquotedblleft = {500,300},
5027 <blg>              \textquoteleft  = {400,400}, \textquoteright = {400,400},
5028 <blg>              \textquotedbl  = {300,300}, \textquotedblleft = {300,300},
5029 <blg>              \textquotedblright = {300,300}, \quotedblbase = {200,600},
5030 <m-t|ptm>          \quotesinglbase = {300,700}, \quotedblbase = {400,500},
5031 <cmr>              \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5032 <bch|pmn>          \quotesinglbase = {200,500}, \quotedblbase = {150,500},
5033 <pad|ppl>          \quotesinglbase = {500,500}, \quotedblbase = {400,400},
5034 <ugm>              \quotesinglbase = {300,700}, \quotedblbase = {300,500},
5035 <m-t|ppl|ptm>      \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5036 <bch|pmn>          \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5037 <cmr>              \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5038 <pad>              \guilsinglleft = {500,400}, \guilsinglright = {300,500},
5039 <ugm>              \guilsinglleft = {400,400}, \guilsinglright = {300,600},
5040 <m-t|ppl>          \guillemotleft  = {300,300}, \guillemotright = {300,300},
5041 <bch|pmn>          \guillemotleft  = {200,300}, \guillemotright = {150,400},
5042 <cmr>              \guillemotleft  = {400,100}, \guillemotright = {200,300},
5043 <pad>              \guillemotleft  = {300,300}, \guillemotright = {200,400},
5044 <ptm>              \guillemotleft  = {300,400}, \guillemotright = {200,400},
5045 <ugm>              \guillemotleft  = {300,400}, \guillemotright = {300,400},
5046 <m-t|pad|ppl|ugm> \textexclamdown = {100,  }, \textquestiondown = {200,  },
5047 <cmr|ptm>          \textexclamdown = {200,  }, \textquestiondown = {200,  },
5048 <pmn>              \textexclamdown = {-50,  }, \textquestiondown = {-50,  },

```

```

5049 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
5050 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
5051 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
5052 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
5053 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
5054 <pmn> \textvisiblespace = {100,100}
5055 }
5056
5057 <*m-t|ptm>
5058 \SetProtrusion
5059 <m-t> [ name = QX-it-default,
5060 <ptm> [ name = ptm-it-QX,
5061 <m-t> load = OT1-it ]
5062 <ptm> load = ptm-it ]
5063 { encoding = {QX},
5064 <ptm> family = {ptm,ptmx,ptmj},
5065 shape = {it,sl} }
5066 {
5067 <ptm> 009 = { , 50}, % fk
5068 {=} = {100,100},
5069 <m-t> \textunderscore = {100,100},
5070 <ptm> \textunderscore = {100,150},
5071 \textbackslash = {100,200},
5072 \quotedblbase = {300,400},
5073 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
5074 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
5075 \textexclamdown = {200, }, \textquestiondown = {200, },
5076 \textbraceleft = {200,100}, \textbraceright = {200,200},
5077 \textless = {100,100}, \textgreater = {100,100},
5078 \textminus = {200,200}, \textdegree = {300,150},
5079 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5080 <ptm> \textregistered = {100,150}, \copyright = {100,150},
5081 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
5082 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
5083 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
5084 <ptm> \textquotelleft = {500,400}, \textquoteright = {500,400},
5085 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
5086 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
5087 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5088 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
5089 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
5090 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
5091 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5092 <ptm> \textperthousand = { ,50}
5093 }
5094
5095 </m-t|ptm>
5096 <*cmr|bch>
5097 \SetProtrusion
5098 <cmr> [ name = cmr-it-T5,
5099 <cmr> load = cmr-it ]
5100 <bch> [ name = bch-it-T5,
5101 <bch> load = bch-it ]
5102 { encoding = T5,
5103 <bch> family = bch,
5104 <cmr> family = cmr,
5105 shape = it }
5106 {
5107 <bch> _ = { ,100},
5108 <cmr> _ = {100,200},
5109 <bch> \textbackslash = {150,150},
5110 <cmr> \textbackslash = {300,300},
5111 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},

```

```

5112 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5113 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5114 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5115 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
5116 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
5117 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
5118 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
5119 <bch> \textless = {100, }, \textgreater = { ,100}
5120 <cmr> \textless = {300,100}, \textgreater = {200,100}
5121 }
5122
5123 </cmr|bch>

```

Slanted is very similar to italic.

```

5124 <*cmr>
5125 \SetProtrusion
5126 [ name = cmr-sl,
5127 load = cmr-it-OT1 ]
5128 { encoding = {OT1,OT4},
5129 family = cmr,
5130 shape = sl }
5131 {
5132 L = { ,50},
5133 f = { ,-50},
5134 - = {300, },
5135 \textendash = {400, }, \textemdash = {300, }
5136 }
5137
5138 \SetProtrusion
5139 [ name = cmr-sl-T1,
5140 load = cmr-it-T1 ]
5141 { encoding = {T1,LY1},
5142 family = cmr,
5143 shape = sl }
5144 {
5145 L = { ,50},
5146 f = { ,-50},
5147 - = {300, },
5148 \textendash = {400, }, \textemdash = {300, }
5149 }
5150
5151 \SetProtrusion
5152 [ name = cmr-sl-T5,
5153 load = cmr-it-T5 ]
5154 { encoding = T5,
5155 family = cmr,
5156 shape = sl }
5157 {
5158 L = { ,50},
5159 f = { ,-50},
5160 - = {300, },
5161 \textendash = {400, }, \textemdash = {300, }
5162 }
5163
5164 \SetProtrusion
5165 [ name = lmr-it-T1,
5166 load = cmr-it-T1 ]
5167 { encoding = {T1,LY1},
5168 family = lmr,
5169 shape = {it,sl} }
5170 {
5171 \textquotedblleft = { ,200}, \textquotedblright = { ,200},
5172 \quotesinglbase = { ,400}, \quotedblbase = { ,500}

```

```
5173 }
5174
```

Oldstyle numerals are slightly different.

```
5175 \SetProtrusion
5176 [ name = cmr(oldstyle)-it,
5177   load = cmr-it-T1 ]
5178 { encoding = T1,
5179   family   = {hfor,cmor},
5180   shape    = {it,sl} }
5181 {
5182   1 = {250, 50},
5183   2 = {150,-100},
5184   3 = {100,-50},
5185   4 = {150,150},
5186   6 = {200,   },
5187   7 = {200, 50},
5188   8 = {150,-50},
5189   9 = {100, 50}
5190 }
5191
5192 </cmr>
5193 < *pmn>
5194 \SetProtrusion
5195 [ name   = pmnx-it,
5196   load   = pmnj-it ]
5197 { encoding = OT1,
5198   family   = pmnx,
5199   shape    = {it,sl} }
5200 {
5201   1 = {100,150}
5202 }
5203
5204 \SetProtrusion
5205 [ name   = pmnx-it-T1,
5206   load   = pmnj-it-T1 ]
5207 { encoding = {T1,LY1},
5208   family   = pmnx,
5209   shape    = {it,sl} }
5210 {
5211   1 = {100,150}
5212 }
5213
5214 </pmn>
5215 < *ptm>
5216 \SetProtrusion
5217 [ name   = ptm-it-LY1,
5218   load   = ptm-it-T1 ]
5219 { encoding = {LY1},
5220   family   = {ptm,ptmx,ptmj},
5221   shape    = {it,sl} }
5222 {
5223   -                               = {100,100},
5224   \texttrademark                 = {100,100},
5225   \textregistered                 = {100,100},
5226   \textcopyright                 = {100,100},
5227   \textdegree                    = {300,100},
5228   \textminus                     = {200,200},
5229   \textellipsis                  = {100,200},
5230   \texteuro                      = {   ,   },
5231   \textcent                      = {100,100},
5232   \textquotesingle               = {500,   },
5233   \textflorin                   = {100, 70},
```

```

5234 \textdagger           = {150,150},
5235 \textdaggerdbl        = {100,100},
5236 \textbullet           = {150,150},
5237 \textonesuperior      = {150,100},
5238 \texttwosuperior      = {150, 50},
5239 \textthreesuperior    = {150, 50},
5240 \textparagraph         = {100,  },
5241 \textperiodcentered    = {500,300},
5242 \textonequarter        = { 50,  },
5243 \textonehalf           = { 50,  },
5244 \textplusminus         = {100,100},
5245 \textmultiply          = {150,150},
5246 \textdivide           = {150,150}
5247 }
5248
5249 </ptm>

```

15.8.3 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).

```

5250 <*(big|ugm)>
5251 \SetProtrusion
5252 <m-t> [ name      = OT1-sc,
5253 <bch> [ name      = bch-sc,
5254 <cmr> [ name      = cmr-sc-OT1,
5255 <pad> [ name      = pad-sc,
5256 <pmn> [ name      = pmnj-sc,
5257 <ppl> [ name      = ppl-sc,
5258 <ptm> [ name      = ptm-sc,
5259 <m-t>   load      = default ]
5260 <bch>   load      = bch-default ]
5261 <cmr>   load      = cmr-OT1 ]
5262 <pad>   load      = pad-default ]
5263 <pmn>   load      = pmnj-default ]
5264 <ppl>   load      = ppl-default ]
5265 <ptm>   load      = ptm-default ]
5266 <m-t|bch|pad|pmn> { encoding = OT1,
5267 <cmr|ppl|ptm> { encoding = {OT1,OT4},
5268 <bch>   family    = bch,
5269 <cmr>   family    = cmr,
5270 <pad>   family    = {pad,padx,padj},
5271 <pmn>   family    = pmnj,
5272 <ppl>   family    = {ppl,pplx,pplj},
5273 <ptm>   family    = {ptm,ptmx,ptmj},
5274   shape  = sc }
5275 {
5276   a = {50,50},
5277 <cmr|pad|ppl|ptm> \ae = {50,  },
5278 <bch|pmn>         c = {50,  },
5279 <bch|pad|pmn>     d = {  ,50},
5280 <m-t|bch|cmr|pad|pmn|ptm> f = {  ,50},
5281 <bch|pad|pmn>     g = {50,  },
5282 <m-t|cmr|pad|pmn|ppl|ptm> j = {50,  },
5283 <bch>             j = {100,  },
5284 <m-t|bch|cmr|pad|pmn|ppl> l = {  ,50},
5285 <ptm>             l = {  ,80},
5286 <m-t|bch|cmr|pad|pmn|ppl> 013 = {  ,50}, % fl
5287 <ptm>             013 = {  ,80}, % fl
5288 <bch|pad|pmn>     o = {50,50},

```

```

5289 <pad|pmn> \oe = {50, },
5290 <ppl> p = { 0, 0},
5291 <bch|pad|pmn> q = {50,70},
5292 <ppl> q = { 0, },
5293 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5294 t = {50,50},
5295 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5296 <ptm> y = {80,80}
5297 }
5298
5299 \SetProtrusion
5300 <m-t> [ name = T1-sc,
5301 <bch> [ name = bch-sc-T1,
5302 <cmr> [ name = cmr-sc-T1,
5303 <pad> [ name = pad-sc-T1,
5304 <pmn> [ name = pmnj-sc-T1,
5305 <ppl> [ name = ppl-sc-T1,
5306 <ptm> [ name = ptm-sc-T1,
5307 <m-t> load = T1-default ]
5308 <bch> load = bch-T1 ]
5309 <cmr> load = cmr-T1 ]
5310 <pad> load = pad-T1 ]
5311 <pmn> load = pmnj-T1 ]
5312 <ppl> load = ppl-T1 ]
5313 <ptm> load = ptm-T1 ]
5314 { encoding = {T1,LY1},
5315 <bch> family = bch,
5316 <cmr> family = cmr,
5317 <pad> family = {pad,padx,padj},
5318 <pmn> family = pmnj,
5319 <ppl> family = {ppl,pplx,pplj},
5320 <ptm> family = {ptm,ptmx,ptmj},
5321 shape = sc }
5322 {
5323 a = {50,50},
5324 <cmr|pad|ppl|ptm> \ae = {50, },
5325 <bch|pmn> c = {50, },
5326 <bch|pad|pmn> d = { ,50},
5327 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
5328 <bch|pad|pmn> g = {50, },
5329 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
5330 <bch> j = {100, },
5331 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
5332 <ptm> l = { ,80},
5333 <m-t|bch|cmr|pad|pmn|ppl> 029 = { ,50}, % fl
5334 <ptm> 029 = { ,80}, % fl
5335 <bch|pad|pmn> o = {50,50},
5336 <bch|pad|pmn> \oe = {50, },
5337 <ppl> p = { 0, 0},
5338 <bch|pad|pmn> q = {50,70},
5339 <ppl> q = { 0, },
5340 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5341 t = {50,50},
5342 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5343 <ptm> y = {80,80}
5344 }
5345
5346 <!(blg|ugm)>
5347 <*m-t>
5348 \SetProtrusion
5349 [ name = QX-sc,
5350 load = QX-default ]
5351 { encoding = QX,

```



```

5352     shape    = sc }
5353     {
5354         a = {50,50},
5355         f = { ,50},
5356         j = {50, },
5357         l = { ,50},
5358         013 = { ,50}, % fl
5359         r = { , 0},
5360         t = {50,50},
5361         y = {50,50}
5362     }
5363
5364 </m-t>
5365 <*cmr|bch>
5366 \SetProtrusion
5367 <bch> [ name      = bch-sc-T5,
5368 <bch>    load      = bch-T5 ]
5369 <cmr> [ name      = cmr-sc-T5,
5370 <cmr>    load      = cmr-T5 ]
5371     { encoding = T5,
5372 <bch>        family = bch,
5373 <cmr>        family = cmr,
5374         shape    = sc }
5375     {
5376         a = {50,50},
5377 <bch>        c = {50, },
5378 <bch>        d = { ,50},
5379         f = { ,50},
5380 <bch>        g = {50, },
5381 <bch>        j = {100, },
5382 <cmr>        j = {50, },
5383         l = { ,50},
5384 <bch>        o = {50,50},
5385 <bch>        q = { 0, },
5386 <cmr>        r = { , 0},
5387         t = {50,50},
5388         y = {50,50}
5389     }
5390
5391 </cmr|bch>
5392 <*pmn>
5393 \SetProtrusion
5394 [ name      = pmnx-sc,
5395   load      = pmnj-sc ]
5396 { encoding = OT1,
5397   family    = pmnx,
5398   shape     = sc }
5399 {
5400     1 = {230,180}
5401 }
5402
5403 \SetProtrusion
5404 [ name      = pmnx-sc-T1,
5405   load      = pmnj-sc-T1 ]
5406 { encoding = {T1,LY1},
5407   family    = pmnx,
5408   shape     = sc }
5409 {
5410     1 = {230,180}
5411 }
5412

```

15.8.4 Italic small caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's `fontinstallationguide` suggests `si`.

```

5413 \SetProtrusion
5414 [ name      = pmnj-scit,
5415   load      = pmnj-it  ]
5416 { encoding = OT1,
5417   family   = pmnj,
5418   shape     = {scit,si} }
5419 {
5420   a = {50, },
5421   \ae = { , -50},
5422   b = {20, -50},
5423   c = {50, -50},
5424   d = {20, 0},
5425   e = {20, -50},
5426   f = {10, 0},
5427   012 = {10, -50}, % fi
5428   013 = {10, -50}, % fl
5429   014 = {10, -50}, % ffi
5430   015 = {10, -50}, % ffl
5431   g = {50, -50},
5432   i = {20, -50},
5433   j = {20, 0},
5434   k = {20, },
5435   l = {20, 50},
5436   m = { , -30},
5437   n = { , -30},
5438   o = {50, },
5439   \oe = {50, -50},
5440   p = {20, -50},
5441   q = {50, },
5442   r = {20, 0},
5443   s = {20, -30},
5444   t = {70, },
5445   u = {50, -50},
5446   v = {100, },
5447   w = {100, },
5448   y = {50, },
5449   z = { , -50}
5450 }
5451
5452 \SetProtrusion
5453 [ name      = pmnj-scit-T1,
5454   load      = pmnj-it-T1  ]
5455 { encoding = {T1,LY1},
5456   family   = pmnj,
5457   shape     = {scit,si}   }
5458 {
5459   a = {50, },
5460   \ae = { , -50},
5461   b = {20, -50},
5462   c = {50, -50},
5463   d = {20, 0},
5464   e = {20, -50},
5465   f = {10, 0},
5466   028 = {10, -50}, % fi
5467   029 = {10, -50}, % fl
5468   030 = {10, -50}, % ffi
5469   031 = {10, -50}, % ffl
5470   g = {50, -50},

```

```

5471     i = {20,-50},
5472 188 = {20, 0}, % ij
5473     j = {20, 0},
5474     k = {20, },
5475     l = {20,50},
5476     m = { , -30},
5477     n = { , -30},
5478     o = {50, },
5479 \oe = {50,-50},
5480     p = {20,-50},
5481     q = {50, },
5482     r = {20, 0},
5483     s = {20,-30},
5484     t = {70, },
5485     u = {50,-50},
5486     v = {100, },
5487     w = {100, },
5488     y = {50, },
5489     z = { , -50}
5490 }
5491
5492 \SetProtrusion
5493 [ name      = pmnx-scit,
5494   load      = pmnj-scit ]
5495 { encoding = OT1,
5496   family   = pmnx,
5497   shape     = {scit,si} }
5498 {
5499   1 = {100,150}
5500 }
5501
5502 \SetProtrusion
5503 [ name      = pmnx-scit-T1,
5504   load      = pmnj-scit-T1 ]
5505 { encoding = {T1,LY1},
5506   family   = pmnx,
5507   shape     = {scit,si} }
5508 {
5509   1 = {100,150}
5510 }
5511
5512 </pmn>

```

15.8.5 Text companion

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

```

5513 \SetProtrusion
5514 <m-t> [ name      = textcomp ]
5515 <bch> [ name      = bch-textcomp ]
5516 <blg> [ name      = blg-textcomp ]
5517 <cmr> [ name      = cmr-textcomp ]
5518 <pad> [ name      = pad-textcomp ]
5519 <pmn> [ name      = pmn-textcomp ]
5520 <ppl> [ name      = ppl-textcomp ]
5521 <ptm> [ name      = ptm-textcomp ]
5522 <ugm> [ name      = ugm-textcomp ]
5523 <m-t> { encoding = TS1 }
5524 <!m-t> { encoding = TS1,
5525 <bch>   family   = bch }
5526 <blg>   family   = blg }

```

```

5527 <cmr>      family    = cmr }
5528 <pad>      family    = {pad,padx,padj} }
5529 <pmn>      family    = {pmnx,pmnj} }
5530 <ppl>      family    = {ppl,pplx,pplj} }
5531 <ptm>      family    = {ptm,ptmx,ptmj} }
5532 <ugm>      family    = ugm }
5533 {
5534 <blg>      \textquotestraightbase    = {400,500},
5535 <cmr>      \textquotestraightbase    = {300,300},
5536 <pad|pmn>  \textquotestraightbase    = {400,400},
5537 <blg>      \textquotestraightdblbase = {300,400},
5538 <cmr|pmn>  \textquotestraightdblbase = {300,300},
5539 <pad>      \textquotestraightdblbase = {400,400},
5540 <bch|cmr|pad|pmn|ugm> \texttwelveudash    = {200,200},
5541 <bch|cmr|pad|pmn>    \textthreequartersemdash = {150,150},
5542 <ugm>      \textthreequartersemdash = {200,200},
5543 <blg>      \textquotesingle          = {500,600},
5544 <cmr|pmn>  \textquotesingle          = {300,400},
5545 <pad>      \textquotesingle          = {400,500},
5546 <ptm>      \textquotesingle          = {500,500},
5547 <ugm>      \textquotesingle          = {300,500},
5548 <bch|cmr|pmn> \textasteriskcentered    = {200,300},
5549 <blg>      \textasteriskcentered    = {150,200},
5550 <pad>      \textasteriskcentered    = {300,300},
5551 <ugm>      \textasteriskcentered    = {100,200},
5552 <pmn>      \textfractionsolidus      = {-200,-200},
5553 <cmr>      \textoneoldstyle          = {100,100},
5554 <pmn>      \textoneoldstyle          = {  , 50},
5555 <cmr>      \textthreeoldstyle         = {  , 50},
5556 <pad|pmn>  \textthreeoldstyle         = { 50,  },
5557 <cmr>      \textfouroldstyle          = { 50, 50},
5558 <pad|pmn>  \textfouroldstyle          = { 50,  },
5559 <cmr|pad|pmn> \textsevenoldstyle       = { 50, 80},
5560 <cmr>      \textlangle               = {400,  },
5561 <cmr>      \textrangle               = {  ,400},
5562 <m-t|bch|pmn|ptm> \textminus          = {200,200},
5563 <cmr|pad|ppl> \textminus          = {300,300},
5564 <blg|ugm> \textminus          = {250,300},
5565 <bch|pad|pmn> \textlbrackdbl         = {100,  },
5566 <blg>      \textlbrackdbl           = {200,  },
5567 <bch|pad|pmn> \textrbrackdbl         = {  ,100},
5568 <blg>      \textrbrackdbl           = {  ,200},
5569 <pmn>      \textasciigrave          = {200,500},
5570 <bch|blg|cmr|pad|pmn> \texttildelow    = {200,250},
5571 <pmn>      \textasciibreve          = {300,400},
5572 <pmn>      \textasciicaron          = {300,400},
5573 <pmn>      \textacutedbl            = {200,300},
5574 <pmn>      \textgravedbl            = {150,300},
5575 <bch|pmn|ugm> \textdagger           = { 80, 80},
5576 <blg>      \textdagger              = {200,200},
5577 <cmr|pad> \textdagger              = {100,100},
5578 <ptm>      \textdagger              = {150,150},
5579 <blg>      \textdaggerdbl           = {150,150},
5580 <cmr|pad|pmn> \textdaggerdbl         = { 80, 80},
5581 <ptm>      \textdaggerdbl           = {100,100},
5582 <bch>      \textbardbl              = {100,100},
5583 <blg|ugm> \textbardbl              = {150,150},
5584 <bch>      \textbullet              = {200,200},
5585 <blg>      \textbullet              = {400,500},
5586 <cmr|pad|pmn> \textbullet            = {  ,100},
5587 <ptm>      \textbullet              = {150,150},
5588 <ugm>      \textbullet              = { 50,100},
5589 <bch|cmr|pmn> \textcelsius          = { 50,  },

```

```

5590 <pad> \textcelsius = { 80, },
5591 <bch> \textflorin = { 50, 50},
5592 <blg> \textflorin = {100,100},
5593 <pad|ugm> \textflorin = { ,100},
5594 <pmn> \textflorin = { 50,100},
5595 <ptm> \textflorin = { 50, 70},
5596 <cmr> \textcolonmonetary = { , 50},
5597 <pad|pmn> \textcolonmonetary = { 50, },
5598 <pmn> \textinterrobang = { ,100},
5599 <pmn> \textinterrobangdown = {100, },
5600 <m-t|pad|ptm> \texttrademark = {100,100},
5601 <bch> \texttrademark = {150,150},
5602 <blg|cmr|ppl> \texttrademark = {200,200},
5603 <pmn> \texttrademark = { 50, 50},
5604 <ugm> \texttrademark = {100,150},
5605 <bch|ugm> \textcent = { 50, },
5606 <ptm> \textcent = {100,100},
5607 <bch> \textsterling = { 50, },
5608 <ugm> \textsterling = { , 50},
5609 <bch> \textbrokenbar = {200,200},
5610 <blg> \textbrokenbar = {250,250},
5611 <ugm> \textbrokenbar = {200,300},
5612 <pmn> \textasciidieresis = {300,400},
5613 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
5614 <pmn> \textcopyright = {100,150},
5615 <ppl> \textcopyright = {200,200},
5616 <bch|cmr|ugm> \textordfeminine = {100,200},
5617 <pad|pmn> \textordfeminine = {200,200},
5618 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
5619 <blg> \textlnot = {200,100},
5620 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
5621 <pmn> \textregistered = { 50,150},
5622 <ppl> \textregistered = {200,200},
5623 <pmn> \textasciimacron = {150,200},
5624 <m-t|ppl|ptm> \textdegree = {300,300},
5625 <bch> \textdegree = {150,200},
5626 <blg|ugm> \textdegree = {200,200},
5627 <cmr|pad> \textdegree = {400,400},
5628 <pmn> \textdegree = {150,400},
5629 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
5630 <blg> \textpm = {100,100},
5631 <ptm> \textpm = { 50, 80},
5632 <bch|blg|ugm> \texttwosuperior = {100,200},
5633 <cmr> \texttwosuperior = { 50,100},
5634 <pad|pmn> \texttwosuperior = {200,200},
5635 <ptm> \texttwosuperior = { 50, 50},
5636 <bch|blg|ugm> \textthreesuperior = {100,200},
5637 <cmr> \textthreesuperior = { 50,100},
5638 <pad|pmn> \textthreesuperior = {200,200},
5639 <ptm> \textthreesuperior = { 50, 50},
5640 <pmn> \textasciiacute = {300,400},
5641 <bch|ugm> \textmu = { ,100},
5642 <bch|pad|pmn> \textparagraph = { ,100},
5643 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
5644 <blg> \textperiodcentered = {400,500},
5645 <ptm> \textperiodcentered = {300,300},
5646 <ugm> \textperiodcentered = {200,500},
5647 <bch|blg|ugm> \textonesuperior = {200,300},
5648 <cmr|pad|pmn> \textonesuperior = {200,200},
5649 <ptm> \textonesuperior = {100,100},
5650 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
5651 <blg|cmr> \textordmasculine = {100,200},
5652 <bch|cmr|pmn> \texteuro = {100, },

```

```

5653 <pad>      \texteuro           = { 50,100},
5654 <bch>      \texttimes          = {200,200},
5655 <blg|ptm>  \texttimes          = {100,100},
5656 <cmr>      \texttimes          = {150,250},
5657 <pad>      \texttimes          = {100,150},
5658 <pmn>      \texttimes          = { 70,100},
5659 <ugm>      \texttimes          = {200,300},
5660 <bch|pad|pmn> \textdiv         = {150,200}
5661 <blg>      \textdiv           = {100,100}
5662 <cmr>      \textdiv           = {150,250}
5663 <ptm>      \textdiv           = { 50,100},
5664 <ugm>      \textdiv           = {200,300},
5665 <ptm>      \textperthousand    = {   ,50}
5666 <ugm>      \textsection        = {   ,100},
5667 <ugm>      \textonehalf        = { 50,100},
5668 <ugm>      \textonequarter     = { 50,100},
5669 <ugm>      \textthreequarters  = { 50,100},
5670 <ugm>      \textsurd           = {   ,100}

```

Remaining slots in the source file.

```

5671   }
5672
5673 <*cmr|pad|pmn|ugm>
5674 \SetProtrusion
5675 <cmr> [ name = cmr-textcomp-it ]
5676 <pad> [ name = pad-textcomp-it ]
5677 <pmn> [ name = pmn-textcomp-it ]
5678 <ugm> [ name = ugm-textcomp-it ]
5679 { encoding = TS1,
5680 <cmr> family = cmr,
5681 <pad> family = {pad,padx,padj},
5682 <pmn> family = {pmnx,pmnj},
5683 <ugm> family = ugm,
5684 <!ugm> shape = {it,sl} }
5685 <ugm> shape = it }
5686 {
5687 <cmr> \textquotestraightbase = {300,600},
5688 <pad|pmn> \textquotestraightbase = {400,400},
5689 <cmr> \textquotestraightdblbase = {300,600},
5690 <pad> \textquotestraightdblbase = {300,400},
5691 <pmn> \textquotestraightdblbase = {300,300},
5692 \texttwelveudash = {200,200},
5693 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
5694 <ugm> \textthreequartersemdash = {200,200},
5695 <cmr> \textquotesingle = {600,300},
5696 <pad> \textquotesingle = {800,100},
5697 <pmn> \textquotesingle = {300,200},
5698 <ugm> \textquotesingle = {500,500},
5699 <cmr> \textasteriskcentered = {300,200},
5700 <pad> \textasteriskcentered = {500,100},
5701 <pmn> \textasteriskcentered = {200,300},
5702 <ugm> \textasteriskcentered = {300,150},
5703 <pmn> \textfractionsolidus = {-200,-200},
5704 <cmr> \textoneoldstyle = {100, 50},
5705 <pad> \textoneoldstyle = {100,  },
5706 <pmn> \textoneoldstyle = { 50,  },
5707 <pad> \texttwooldstyle = { 50,  },
5708 <pmn> \texttwooldstyle = {-50,  },
5709 <cmr> \textthreeoldstyle = {100, 50},
5710 <pmn> \textthreeoldstyle = {-100,  },
5711 <cmr> \textfouroldstyle = { 50, 50},
5712 <pad> \textfouroldstyle = { 50,100},
5713 <cmr> \textsevenoldstyle = { 50, 80},

```

5714	<i><pad></i>	<code>\textsevenoldstyle</code>	= { 50, },
5715	<i><pmn></i>	<code>\textsevenoldstyle</code>	= { 20, },
5716	<i><cmr></i>	<code>\textlangle</code>	= {400, },
5717	<i><cmr></i>	<code>\textrangle</code>	= { ,400},
5718	<i><cmr pad></i>	<code>\textminus</code>	= {300,300},
5719	<i><pmn></i>	<code>\textminus</code>	= {200,200},
5720	<i><ugm></i>	<code>\textminus</code>	= {250,300},
5721	<i><pad pmn></i>	<code>\textlbrackdbl</code>	= {100, },
5722	<i><pad pmn></i>	<code>\textrbrackdbl</code>	= { ,100},
5723	<i><pmn></i>	<code>\textasciigrave</code>	= {300,300},
5724	<i><cmr pad pmn></i>	<code>\texttildelow</code>	= {200,250},
5725	<i><pmn></i>	<code>\textasciibreve</code>	= {300,300},
5726	<i><pmn></i>	<code>\textasciicaron</code>	= {300,300},
5727	<i><pmn></i>	<code>\textacutedbl</code>	= {200,300},
5728	<i><pmn></i>	<code>\textgravedbl</code>	= {150,300},
5729	<i><cmr></i>	<code>\textdagger</code>	= {100,100},
5730	<i><pad></i>	<code>\textdagger</code>	= {200,100},
5731	<i><pmn></i>	<code>\textdagger</code>	= { 80, 50},
5732	<i><ugm></i>	<code>\textdagger</code>	= { 80, 80},
5733	<i><cmr pad></i>	<code>\textdaggerdbl</code>	= { 80, 80},
5734	<i><pmn></i>	<code>\textdaggerdbl</code>	= { 80, 50},
5735	<i><ugm></i>	<code>\textbardbl</code>	= {150,150},
5736	<i><cmr></i>	<code>\textbullet</code>	= {200,100},
5737	<i><pad></i>	<code>\textbullet</code>	= {300, },
5738	<i><pmn></i>	<code>\textbullet</code>	= { 30, 70},
5739	<i><ugm></i>	<code>\textbullet</code>	= { 50,100},
5740	<i><cmr></i>	<code>\textcelsius</code>	= {100, },
5741	<i><pad></i>	<code>\textcelsius</code>	= {200, },
5742	<i><pmn></i>	<code>\textcelsius</code>	= { 50,-50},
5743	<i><pad></i>	<code>\textflorin</code>	= {100, },
5744	<i><pmn></i>	<code>\textflorin</code>	= { 50,100},
5745	<i><ugm></i>	<code>\textflorin</code>	= { ,100},
5746	<i><cmr></i>	<code>\textcolonmonetary</code>	= {150, },
5747	<i><pad></i>	<code>\textcolonmonetary</code>	= {100, },
5748	<i><pmn></i>	<code>\textcolonmonetary</code>	= { 50,-50},
5749	<i><cmr pad></i>	<code>\texttrademark</code>	= {200, },
5750	<i><pmn></i>	<code>\texttrademark</code>	= { 50,100},
5751	<i><ugm></i>	<code>\texttrademark</code>	= {150, 50},
5752	<i><ugm></i>	<code>\textcent</code>	= { 50, },
5753	<i><ugm></i>	<code>\textsterling</code>	= { , 50},
5754	<i><ugm></i>	<code>\textbrokenbar</code>	= {200,300},
5755	<i><pmn></i>	<code>\textasciidieresis</code>	= {300,200},
5756	<i><cmr></i>	<code>\textcopyright</code>	= {100, },
5757	<i><pad></i>	<code>\textcopyright</code>	= {200,100},
5758	<i><pmn></i>	<code>\textcopyright</code>	= {100,150},
5759	<i><ugm></i>	<code>\textcopyright</code>	= {300, },
5760	<i><cmr></i>	<code>\textordfeminine</code>	= {100,100},
5761	<i><pmn></i>	<code>\textordfeminine</code>	= {200,200},
5762	<i><ugm></i>	<code>\textordfeminine</code>	= {100,200},
5763	<i><cmr pad></i>	<code>\textlnot</code>	= {300, },
5764	<i><pmn ugm></i>	<code>\textlnot</code>	= {200, },
5765	<i><cmr></i>	<code>\textregistered</code>	= {100, },
5766	<i><pad></i>	<code>\textregistered</code>	= {200,100},
5767	<i><pmn></i>	<code>\textregistered</code>	= { 50,150},
5768	<i><ugm></i>	<code>\textregistered</code>	= {300, },
5769	<i><pmn></i>	<code>\textasciimacron</code>	= {150,200},
5770	<i><cmr pad></i>	<code>\textdegree</code>	= {500,100},
5771	<i><pmn></i>	<code>\textdegree</code>	= {150,150},
5772	<i><ugm></i>	<code>\textdegree</code>	= {300,200},
5773	<i><cmr></i>	<code>\textpm</code>	= {150,100},
5774	<i><pad></i>	<code>\textpm</code>	= {200,150},
5775	<i><pmn ugm></i>	<code>\textpm</code>	= {150,200},
5776	<i><cmr></i>	<code>\textonesuperior</code>	= {400, },

```

5777 <pad>      \textonesuperior      = {300,100},
5778 <pmn>      \textonesuperior      = {200,100},
5779 <ugm>      \textonesuperior      = {300,300},
5780 <cmr>      \texttwosuperior       = {400,  },
5781 <pad>      \texttwosuperior       = {300,  },
5782 <pmn>      \texttwosuperior       = {200,100},
5783 <ugm>      \texttwosuperior       = {300,200},
5784 <cmr>      \textthreesuperior     = {400,  },
5785 <pad>      \textthreesuperior     = {300,  },
5786 <pmn>      \textthreesuperior     = {200,100},
5787 <ugm>      \textthreesuperior     = {300,200},
5788 <ugm>      \textmu                 = {  ,100},
5789 <pmn>      \textasciicute         = {300,200},
5790 <cmr>      \textparagraph          = {200,  },
5791 <pmn>      \textparagraph          = {  ,100},
5792 <cmr>      \textperiodcentered    = {500,500},
5793 <pad|pmn|ugm> \textperiodcentered = {300,400},
5794 <cmr>      \textordmasculine       = {100,100},
5795 <pmn>      \textordmasculine       = {200,200},
5796 <ugm>      \textordmasculine       = {300,200},
5797 <cmr>      \texteuro               = {200,  },
5798 <pad>      \texteuro               = {100,  },
5799 <pmn>      \texteuro               = {100,-50},
5800 <cmr>      \texttimes              = {200,200},
5801 <pad>      \texttimes              = {200,100},
5802 <pmn>      \texttimes              = { 70,100},
5803 <ugm>      \texttimes              = {200,300},
5804 <cmr|pad>  \textdiv               = {200,200}
5805 <pmn>      \textdiv               = {150,200}
5806 <ugm>      \textdiv               = {200,300},
5807 <ugm>      \textsection            = {  ,200},
5808 <ugm>      \textonehalf            = { 50,100},
5809 <ugm>      \textonequarter         = { 50,100},
5810 <ugm>      \textthreequarters     = { 50,100},
5811 <ugm>      \textsurd               = {  ,100}
5812 }
5813
5814 </cmr|pad|pmn|ugm>

```

15.8.6 Computer Modern math

Now to the math symbols for Computer Modern Roman. Definitions have been extracted from `fontmath.ltx`. I did not spend too much time fiddling with these settings, so they can surely be improved.

The math font ‘operators’ (also used for the `\mathrm` and `\mathbf` alphabets) is OT1/cmr, which we’ve already set up above. It’s declared as:

```

\DeclareSymbolFont{operators} {OT1}{cmr} {m}{n}
\SetSymbolFont{operators}{bold}{OT1}{cmr} {bx}{n}

```

`\mathit` (OT1/cmr/m/it) is also already set up.

There are (for the moment) no settings for `\mathsf` and `\mathtt`.

Math font ‘letters’ (also used as `\mathnormal`) is declared as:

```

\DeclareSymbolFont{letters} {OML}{cmm} {m}{it}
\SetSymbolFont{letters} {bold}{OML}{cmm} {b}{it}

```

```

5815 <*cmr>
5816 \SetProtrusion
5817 [ name = cmr-math-letters ]

```



```

5818 { encoding = OML,
5819     family   = cmm,
5820     series    = {m,b},
5821     shape     = it   }
5822 {
5823     A = {100, 50}, % \mathnormal
5824     B = { 50,   },
5825     C = { 50,   },
5826     D = { 50, 50},
5827     E = { 50,   },
5828     F = {100, 50},
5829     G = { 50, 50},
5830     H = { 50, 50},
5831     I = { 50, 50},
5832     J = {150, 50},
5833     K = { 50,100},
5834     L = { 50, 50},
5835     M = { 50,   },
5836     N = { 50,   },
5837     O = { 50,   },
5838     P = { 50,   },
5839     Q = { 50, 50},
5840     R = { 50,   },
5841     S = { 50,   },
5842     T = { 50,100},
5843     U = { 50, 50},
5844     V = {100,100},
5845     W = { 50,100},
5846     X = { 50,100},
5847     Y = {100,100},
5848     f = {100,100},
5849     h = {   ,100},
5850     i = {   , 50},
5851     j = {   , 50},
5852     k = {   , 50},
5853     r = {   , 50},
5854     v = {   , 50},
5855     w = {   , 50},
5856     x = {   , 50},
5857     "0B = { 50,100}, % \alpha
5858     "0C = { 50, 50}, % \beta
5859     "0D = {200,150}, % \gamma
5860     "0E = { 50, 50}, % \delta
5861     "0F = { 50, 50}, % \epsilon
5862     "10 = { 50,150}, % \zeta
5863     "12 = { 50,   }, % \theta
5864     "13 = {   ,100}, % \iota
5865     "14 = {   ,100}, % \kappa
5866     "15 = {100, 50}, % \lambda
5867     "16 = {   , 50}, % \mu
5868     "17 = {   , 50}, % \nu
5869     "18 = {   , 50}, % \xi
5870     "19 = { 50,100}, % \pi
5871     "1A = { 50, 50}, % \rho
5872     "1B = {   ,150}, % \sigma
5873     "1C = { 50,150}, % \tau
5874     "1D = { 50, 50}, % \upsilon
5875     "1F = { 50,100}, % \chi
5876     "20 = { 50, 50}, % \psi
5877     "21 = {   , 50}, % \omega
5878     "22 = {   , 50}, % \varepsilon
5879     "23 = {   , 50}, % \vartheta
5880     "24 = {   , 50}, % \varpi

```

```

5881 "25 = {100, }, % \varrho
5882 "26 = {100,100}, % \varsigma
5883 "27 = { 50, 50}, % \varphi
5884 "28 = {100,100}, % \leftharpoonup
5885 "29 = {100,100}, % \leftharpoondown
5886 "2A = {100,100}, % \rightharpoonup
5887 "2B = {100,100}, % \rightharpoondown
5888 "2C = {300,200}, % \lhook
5889 "2D = {200,300}, % \rhook
5890 "2E = { ,100}, % \triangleright
5891 "2F = {100, }, % \triangleleft
5892 "3A = { ,500}, % ., \ldotp
5893 "3B = { ,500}, % ,
5894 "3C = {200,100}, % <
5895 "3D = {300,400}, % /
5896 "3E = {100,200}, % >
5897 "3F = {200,200}, % \star
5898 "5B = { ,100}, % \flat
5899 "5E = {200,200}, % \smile
5900 "5F = {200,200}, % \frown
5901 "7C = {100, }, % \jmath
5902 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

5903 }
5904

```

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}

```

```

5905 \SetProtrusion
5906 [ name = cmr-math-symbols ]
5907 { encoding = OMS,
5908   family = cmsy,
5909   series = {m,b},
5910   shape = n }
5911 {
5912   A = {150, 50}, % \mathcal
5913   C = { ,100},
5914   D = { , 50},
5915   F = { 50,150},
5916   I = { ,100},
5917   J = {100,150},
5918   K = { ,100},
5919   L = {100, },
5920   M = { 50, 50},
5921   N = { 50,100},
5922   P = { , 50},
5923   Q = { 50, },
5924   R = { , 50},
5925   T = { 50,150},
5926   V = { 50, 50},
5927   W = { , 50},
5928   X = {100,100},
5929   Y = {100, },
5930   Z = {100,150},
5931   "00 = {300,300}, % -
5932   "01 = { ,700}, % \cdot, \cdotp
5933   "02 = {150,250}, % \times
5934   "03 = {150,250}, % *, \ast
5935   "04 = {200,300}, % \div

```

```

5936 "05 = {150,250}, % \diamond
5937 "06 = {200,200}, % \pm
5938 "07 = {200,200}, % \mp
5939 "08 = {100,100}, % \oplus
5940 "09 = {100,100}, % \ominus
5941 "0A = {100,100}, % \otimes
5942 "0B = {100,100}, % \oslash
5943 "0C = {100,100}, % \odot
5944 "0D = {100,100}, % \bigcirc
5945 "0E = {100,100}, % \circ
5946 "0F = {100,100}, % \bullet
5947 "10 = {100,100}, % \asymp
5948 "11 = {100,100}, % \equiv
5949 "12 = {200,100}, % \subseteq
5950 "13 = {100,200}, % \supseteq
5951 "14 = {200,100}, % \leq
5952 "15 = {100,200}, % \geq
5953 "16 = {200,100}, % \preceq
5954 "17 = {100,200}, % \succeq
5955 "18 = {200,200}, % \sim
5956 "19 = {150,150}, % \approx
5957 "1A = {200,100}, % \subset
5958 "1B = {100,200}, % \supset
5959 "1C = {200,100}, % \ll
5960 "1D = {100,200}, % \gg
5961 "1E = {300,100}, % \prec
5962 "1F = {100,300}, % \succ
5963 "20 = {100,200}, % \leftarrow
5964 "21 = {200,100}, % \rightarrow
5965 "22 = {100,100}, % \uparrow
5966 "23 = {100,100}, % \downarrow
5967 "24 = {100,100}, % \leftrightarrows
5968 "25 = {100,100}, % \nearrow
5969 "26 = {100,100}, % \searrow
5970 "27 = {100,100}, % \simeq
5971 "28 = {100,100}, % \Leftarrow
5972 "29 = {100,100}, % \Rightarrow
5973 "2A = {100,100}, % \Uparrow
5974 "2B = {100,100}, % \Downarrow
5975 "2C = {100,100}, % \Leftrightarrow
5976 "2D = {100,100}, % \nrightarrow
5977 "2E = {100,100}, % \swarrow
5978 "2F = { ,100}, % \propto
5979 "30 = { ,400}, % \prime
5980 "31 = {100,100}, % \infty
5981 "32 = {150,100}, % \ln
5982 "33 = {100,150}, % \ni
5983 "34 = {100,100}, % \triangle, \bigtriangleup
5984 "35 = {100,100}, % \bigtriangledown
5985 "38 = { ,100}, % \forall
5986 "39 = {100, }, % \exists
5987 "3A = {200, }, % \neg
5988 "3E = {200,200}, % \top
5989 "3F = {200,200}, % \bot, \perp
5990 "5E = {100,200}, % \wedge
5991 "5F = {100,200}, % \vee
5992 "60 = { ,300}, % \vdash
5993 "61 = {300, }, % \dashv
5994 "62 = {100,100}, % \lfloor
5995 "63 = {100,100}, % \rfloor
5996 "64 = {100,100}, % \lceil
5997 "65 = {100,100}, % \rceil
5998 "66 = {150, }, % \lbrace

```

```

5999 "67 = { ,150}, % \rbrace
6000 "68 = {400, }, % \langle
6001 "69 = { ,400}, % \rangle
6002 "6C = {100,100}, % \updownarrow
6003 "6D = {100,100}, % \Updownarrow
6004 "6E = {100,300}, % \, \backslash, \setminus
6005 "72 = {100,100}, % \nabla
6006 "79 = {200,200}, % \dagger
6007 "7A = {100,100}, % \ddagger
6008 "7B = {100, }, % \mathparagraph
6009 "7C = {100,100}, % \clubsuit
6010 "7D = {100,100}, % \diamondsuit
6011 "7E = {100,100}, % \heartsuit
6012 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

6013 }
6014

```

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}
```

```

6015 </cmr>
6016 </cfg-t>

```

15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
6017 <*cfg-u>
```

Symbol font 'a'.

```

6018 <*msa>
6019 \SetProtrusion
6020 [ name = AMS-a ]
6021 { encoding = U,
6022   family = msa }
6023 {
6024   "05 = {150,250}, % \centerdot
6025   "06 = {100,100}, % \lozenge
6026   "07 = { 50, 50}, % \blacklozenge
6027   "08 = { 50, 50}, % \circlearrowright
6028   "09 = { 50, 50}, % \circlearrowleft
6029   "0A = {100,100}, % \rightleftharpoons
6030   "0B = {100,100}, % \leftrightharpoons
6031   "0D = {-50,200}, % \Vdash
6032   "0E = {-50,200}, % \Vvdash
6033   "0F = {-70,150}, % \vdash
6034   "10 = {100,150}, % \twoheadrightarrow
6035   "11 = {100,150}, % \twoheadleftarrow
6036   "12 = { 50,100}, % \leftleftarrows
6037   "13 = { 50, 80}, % \rightrightarrows
6038   "14 = {120,120}, % \upuparrows
6039   "15 = {120,120}, % \downdownarrows
6040   "16 = {200,200}, % \upharpoonright
6041   "17 = {200,200}, % \downharpoonright
6042   "18 = {200,200}, % \upharpoonleft
6043   "19 = {200,200}, % \downharpoonleft
6044   "1A = { 80,100}, % \rightarrowtail
6045   "1B = { 80,100}, % \leftarrowtail

```

```

6046 "1C = { 50, 50}, % \leftrightharrows
6047 "1D = { 50, 50}, % \rightleftarrows
6048 "1E = {250, }, % \Lsh
6049 "1F = { ,250}, % \Rsh
6050 "20 = {100,100}, % \rightsquigarrow
6051 "21 = {100,100}, % \leftrightsquigarrow
6052 "22 = {100, 50}, % \looparrowleft
6053 "23 = { 50,100}, % \looparrowright
6054 "24 = { 50, 80}, % \circeq
6055 "25 = { ,100}, % \succsim
6056 "26 = { ,100}, % \gtrsim
6057 "27 = { ,100}, % \gtrapprox
6058 "28 = {150, 50}, % \multimap
6059 "2B = {100,150}, % \doteqdot
6060 "2C = {100,150}, % \triangleq
6061 "2D = {100, 50}, % \precsim
6062 "2E = {100, 50}, % \lesssim
6063 "2F = { 50, 50}, % \lessapprox
6064 "30 = {100, 50}, % \eqslantless
6065 "31 = { 50, 50}, % \eqslantgtr
6066 "32 = {100, 50}, % \curlyeqprec
6067 "33 = { 50,100}, % \curlyeqsucc
6068 "34 = {100, 50}, % \preccurlyeq
6069 "36 = { 50, }, % \leqslant
6070 "38 = { , 50}, % \backprime
6071 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
6072 "3C = { 50,100}, % \succcurlyeq
6073 "3E = { , 50}, % \geqslant
6074 "40 = { , 50}, % \sqsubset
6075 "41 = { 50, }, % \sqsupset
6076 "42 = { ,150}, % \vartriangleright, \rhd
6077 "43 = {150, }, % \vartriangleleft, \lhd
6078 "44 = { ,100}, % \trianglerighteq, \unrhd
6079 "45 = {100, }, % \trianglelefteq, \unlhd
6080 "46 = {100,100}, % \bigstar
6081 "48 = { 50, 50}, % \blacktriangledown
6082 "49 = { ,100}, % \blacktriangleright
6083 "4A = {100, }, % \blacktriangleleft
6084 "4B = { ,150}, % \dashrightarrow (the arrow)
6085 "4C = {150, }, % \dashleftarrow
6086 "4D = { 50, 50}, % \vartriangle
6087 "4E = { 50, 50}, % \blacktriangle
6088 "4F = { 50, 50}, % \triangledown
6089 "50 = { 50, 50}, % \eqcirc
6090 "56 = { ,150}, % \Rrightarrow
6091 "57 = {150, }, % \Lleftarrow
6092 "58 = {100,300}, % \checkmark
6093 "5C = { 50, 50}, % \angle
6094 "5D = { 50, 50}, % \measuredangle
6095 "5E = { 50, 50}, % \sphericalangle
6096 "5F = { , 50}, % \varpropto
6097 "60 = {100,100}, % \smallsmile
6098 "61 = {100,100}, % \smallfrown
6099 "62 = { 50, }, % \Subset
6100 "63 = { , 50}, % \Supset
6101 "66 = {150,150}, % \curlywedge
6102 "67 = {150,150}, % \curlyvee
6103 "68 = { 50,150}, % \leftthreetimes
6104 "69 = {100, 50}, % \rightthreetimes
6105 "6C = { 50, 50}, % \bumpeq
6106 "6D = { 50, 50}, % \Bumpeq
6107 "6E = {100, }, % \lll
6108 "6F = { ,100}, % \ggg

```

```

6109 "70 = { 50,100}, % \ulcorner
6110 "71 = {100, 50}, % \urcorner
6111 "75 = {150,200}, % \dotplus
6112 "76 = { 50,100}, % \backsim
6113 "78 = { 50,100}, % \llcorner
6114 "79 = {100, 50}, % \lrcorner
6115 "7C = {100,100}, % \intercal
6116 "7D = { 50, 50}, % \circledcirc
6117 "7E = { 50, 50}, % \circledast
6118 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

6119 }
6120
6121 </msa>

```

Symbol font ‘b’.

```

6122 <*msb>
6123 \SetProtrusion
6124 [ name = AMS-b ]
6125 { encoding = U,
6126   family = msb }
6127 {
6128   A = { 50, 50}, % \mathbb
6129   C = { 50, 50},
6130   G = { , 50},
6131   L = { , 50},
6132   P = { , 50},
6133   R = { , 50},
6134   T = { , 50},
6135   V = { 50, 50},
6136   X = { 50, 50},
6137   Y = { 50, 50},
6138   "00 = { 50, 50}, % \lvertneqq
6139   "01 = { 50, 50}, % \gvertneqq
6140   "02 = { 50, 50}, % \nleq
6141   "03 = { 50, 50}, % \ngeq
6142   "04 = {100, 50}, % \nless
6143   "05 = { 50,150}, % \ngtr
6144   "06 = {100, 50}, % \nprec
6145   "07 = { 50,150}, % \nsucc
6146   "08 = { 50, 50}, % \lneqq
6147   "09 = { 50, 50}, % \gneqq
6148   "0A = {100,100}, % \nleqslant
6149   "0B = {100,100}, % \ngeqslant
6150   "0C = {100, 50}, % \lneq
6151   "0D = { 50,100}, % \gneq
6152   "0E = {100, 50}, % \npreceq
6153   "0F = { 50,100}, % \nsucceq
6154   "10 = { 50, }, % \precnsim
6155   "11 = { 50, 50}, % \succnsim
6156   "12 = { 50, 50}, % \lnsim
6157   "13 = { 50, 50}, % \gnsim
6158   "14 = { 50, 50}, % \nleqq
6159   "15 = { 50, 50}, % \ngeqq
6160   "16 = { 50, 50}, % \precneqq
6161   "17 = { 50, 50}, % \succneqq
6162   "18 = { 50, 50}, % \precnapprox
6163   "19 = { 50, 50}, % \succnapprox
6164   "1A = { 50, 50}, % \lnapprox
6165   "1B = { 50, 50}, % \gnapprox
6166   "1C = {150,200}, % \nsim
6167   "1D = { 50, 50}, % \ncong

```

```

6168 "1E = {100,150}, % \diagup
6169 "1F = {100,150}, % \diagdown
6170 "20 = {100, 50}, % \varsubsetneq
6171 "21 = { 50,100}, % \varsupsetneq
6172 "22 = {100, 50}, % \subsetneqq
6173 "23 = { 50,100}, % \supsetneqq
6174 "24 = {100, 50}, % \subsetneqq
6175 "25 = { 50,100}, % \supsetneqq
6176 "26 = {100, 50}, % \varsubsetneqq
6177 "27 = { 50,100}, % \varsupsetneqq
6178 "28 = {100, 50}, % \subsetneq
6179 "29 = { 50,100}, % \supsetneq
6180 "2A = {100, 50}, % \subseteq
6181 "2B = { 50,100}, % \supseteq
6182 "2C = { 50,100}, % \parallel
6183 "2D = {100,150}, % \mid
6184 "2E = {150,150}, % \shortmid
6185 "2F = {100,100}, % \shortparallel
6186 "30 = { ,150}, % \vdash
6187 "31 = { ,150}, % \Vdash
6188 "32 = { ,100}, % \nVDash
6189 "33 = { ,100}, % \nVDash
6190 "34 = { ,100}, % \ntrianglerighteq
6191 "35 = {100, }, % \trianglelefteq
6192 "36 = {100, }, % \triangleleft
6193 "37 = { ,100}, % \triangleright
6194 "38 = {100,200}, % \leftarrow
6195 "39 = {100,200}, % \rightarrow
6196 "3A = {100,100}, % \Leftarrow
6197 "3B = { 50,100}, % \Rightarrow
6198 "3C = {100,100}, % \Leftrightarrow
6199 "3D = {100,200}, % \leftrightharrow
6200 "3E = { 50, 50}, % \divideontimes
6201 "3F = { 50, 50}, % \varnothing
6202 "60 = {200, }, % \Finv
6203 "61 = { , 50}, % \Game
6204 "68 = {100,100}, % \eqsim
6205 "69 = { 50, }, % \beth
6206 "6A = { 50, }, % \gimel
6207 "6B = {150, }, % \daleth
6208 "6C = {200, }, % \lessdot
6209 "6D = { ,200}, % \gtrdot
6210 "6E = {100,200}, % \ltimes
6211 "6F = {150,100}, % \rtimes
6212 "70 = { 50,100}, % \shortmid
6213 "71 = { 50, 50}, % \shortparallel
6214 "72 = {200,300}, % \smallsetminus
6215 "73 = {100,200}, % \thicksim
6216 "74 = { 50,100}, % \thickapprox
6217 "75 = { 50, 50}, % \approx
6218 "76 = { 50,100}, % \succapprox
6219 "77 = { 50, 50}, % \precapprox
6220 "78 = {100,100}, % \curvearrowleft
6221 "79 = { 50,150}, % \curvearrowright
6222 "7A = { 50,200}, % \digamma
6223 "7B = {100, 50}, % \varkappa
6224 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

6225 }
6226
6227 (/msb)

```

15.8.8 Euler

Euler Roman font (package `euler`).

```

6228 < *eur>
6229 \SetProtrusion
6230 [ name = euler ]
6231 { encoding = U,
6232   family = eur }
6233 {
6234   "01 = {100,100},
6235   "03 = {100,150},
6236   "06 = { ,100},
6237   "07 = {100,150},
6238   "08 = {100,100},
6239   "0A = {100,100},
6240   "0B = { , 50},
6241   "0C = { ,100},
6242   "0D = {100,100},
6243   "0E = { ,100},
6244   "0F = {100,100},
6245   "10 = {100,100},
6246   "13 = { ,100},
6247   "14 = { ,100},
6248   "15 = { , 50},
6249   "16 = { , 50},
6250   "17 = { 50,100},
6251   "18 = { 50,100},
6252   "1A = { , 50},
6253   "1B = { , 50},
6254   "1C = { 50,100},
6255   "1D = { 50,100},
6256   "1E = { 50,100},
6257   "1F = { 50,100},
6258   "20 = { , 50},
6259   "21 = { , 50},
6260   "22 = { 50,100},
6261   "24 = { , 50},
6262   "27 = { 50,100},
6263   1 = {100,100},
6264   7 = { 50,100},
6265   "3A = {300,500},
6266   "3B = {200,400},
6267   "3C = {200,100},
6268   "3D = {200,200},
6269   "3E = {100,200},
6270   A = { ,100},
6271   D = { , 50},
6272   J = { 50, },
6273   K = { , 50},
6274   L = { , 50},
6275   Q = { , 50},
6276   T = { 50, },
6277   X = { 50, 50},
6278   Y = { 50, },
6279   h = { , 50},
6280   k = { , 50}
6281 }
6282

```

Extended by the `eulervm` package.

```

6283 \SetProtrusion
6284 [ name = euler-vm,

```



```

6285     load      = euler ]
6286     { encoding = U,
6287     family     = zeur }
6288     {
6289         "28 = {100,200},
6290         "29 = {100,200},
6291         "2A = {100,150},
6292         "2B = {100,150},
6293         "2C = {200,300},
6294         "2D = {200,300},
6295         "2E = {    ,100},
6296         "2F = {100,   },
6297         "3F = {150,150},
6298         "5B = {    ,100},
6299         "5E = {100,100},
6300         "5F = {100,100},
6301         "80 = {    , 50},
6302         "81 = {200,250},
6303         "82 = {100,200}
6304     }
6305
6306 </eur>

```

Euler Script font (eucal).

```

6307 <*eus>
6308 \SetProtrusion
6309     [ name      = euscript ]
6310     { encoding = U,
6311     family     = eus }
6312     {
6313         A = {100,100},
6314         B = { 50,100},
6315         C = { 50, 50},
6316         D = { 50,100},
6317         E = { 50,100},
6318         F = { 50,   },
6319         G = { 50,   },
6320         H = {    ,100},
6321         K = {    , 50},
6322         L = {    ,150},
6323         M = {    , 50},
6324         N = {    , 50},
6325         O = { 50, 50},
6326         P = { 50, 50},
6327         T = {    ,100},
6328         U = {    , 50},
6329         V = { 50, 50},
6330         W = { 50, 50},
6331         X = { 50, 50},
6332         Y = { 50,   },
6333         Z = { 50,100},
6334         "00 = {250,250},
6335         "18 = {200,200},
6336         "3A = {200,150},
6337         "40 = {    ,100},
6338         "5E = {100,100},
6339         "5F = {100,100},
6340         "66 = { 50,   },
6341         "67 = {    , 50},
6342         "6E = {200,200}
6343     }
6344
6345 \SetProtrusion

```

```
6346 [ name      = euscript-vm,
6347     load      = euscript ]
6348 { encoding = U,
6349   family   = zeus  }
6350 {
6351   "01 = {600,600},
6352   "02 = {200,200},
6353   "03 = {200,200},
6354   "04 = {200,200},
6355   "05 = {150,150},
6356   "06 = {200,200},
6357   "07 = {200,200},
6358   "08 = {100,100},
6359   "09 = {100,100},
6360   "0A = {100,100},
6361   "0B = {100,100},
6362   "0C = {100,100},
6363   "0D = {100,100},
6364   "0E = {150,150},
6365   "0F = {100,100},
6366   "10 = {150,150},
6367   "11 = {100,100},
6368   "12 = {150,100},
6369   "13 = {100,150},
6370   "14 = {150,100},
6371   "15 = {100,150},
6372   "16 = {200,100},
6373   "17 = {100,200},
6374   "19 = {150,150},
6375   "1A = {150,100},
6376   "1B = {100,150},
6377   "1C = {100,100},
6378   "1D = {100,100},
6379   "1E = {250,100},
6380   "1F = {100,250},
6381   "20 = {150,200},
6382   "21 = {150,200},
6383   "22 = {150,150},
6384   "23 = {150,150},
6385   "24 = {100,200},
6386   "25 = {150,150},
6387   "26 = {150,150},
6388   "27 = {100,100},
6389   "28 = {100,100},
6390   "29 = {100,150},
6391   "2A = {100,100},
6392   "2B = {100,100},
6393   "2C = {100,100},
6394   "2D = {150,150},
6395   "2E = {150,150},
6396   "2F = {100,100},
6397   "30 = {100,100},
6398   "31 = {100,100},
6399   "32 = {100,100},
6400   "33 = {100,100},
6401   "34 = {100,100},
6402   "35 = {100,100},
6403   "3E = {150,150},
6404   "3F = {150,150},
6405   "60 = {    ,200},
6406   "61 = {200,   },
6407   "62 = {100,100},
6408   "63 = {100,100},
```

```

6409 "64 = {100,100},
6410 "65 = {100,100},
6411 "68 = {300, },
6412 "69 = { ,300},
6413 "6C = {100,100},
6414 "6D = {100,100},
6415 "6F = {100,100},
6416 "72 = {100,100},
6417 "73 = {200,100},
6418 "76 = { ,100},
6419 "77 = {100, },
6420 "78 = { 50, 50},
6421 "79 = {100,100},
6422 "7A = {100,100},
6423 "7D = {150,150},
6424 "7E = {100,100},
6425 "A8 = {100,100},
6426 "A9 = {100,100},
6427 "AB = {200,200},
6428 "BA = { ,200},
6429 "BB = { ,200},
6430 "BD = {200,200},
6431 "DE = {200,200}
6432 }
6433
6434 </eus>

```

Euler Fraktur font (eufrak).

```

6435 <(*euf)
6436 \SetProtrusion
6437 [ name = mathfrak ]
6438 { encoding = U,
6439 family = euf }
6440 {
6441 A = { , 50},
6442 B = { , 50},
6443 C = { 50, 50},
6444 D = { , 80},
6445 E = { 50, },
6446 G = { , 50},
6447 L = { , 80},
6448 O = { , 50},
6449 T = { , 80},
6450 X = { 80, 50},
6451 Z = { 80, 50},
6452 b = { , 50},
6453 c = { , 50},
6454 k = { , 50},
6455 p = { , 50},
6456 q = { 50, },
6457 v = { , 50},
6458 w = { , 50},
6459 x = { , 50},
6460 1 = {100,100},
6461 2 = { 80, 80},
6462 3 = { 80, 50},
6463 4 = { 80, 50},
6464 7 = { 50, 50},
6465 "12 = {500,500},
6466 "13 = {500,500},
6467 ! = { ,200},
6468 ' = {200,300},
6469 ( = {200, },

```

```

6470      ) = { ,200},
6471      * = {200,200},
6472      + = {200,250},
6473      - = {200,200},
6474      {,} = {300,300},
6475      . = {400,400},
6476      {=} = {200,200},
6477      : = { ,200},
6478      ; = { ,200},
6479      ] = { ,200}
6480  }
6481
6482 </euf>
6483 </cfg-u>

```

15.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym¹⁴).

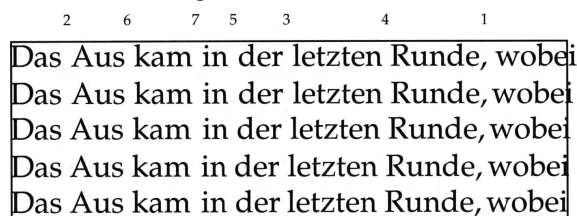
```

6484 <*cfg-e>
6485 \SetProtrusion
6486 <zpeu|euroitc> { encoding = U,
6487 <mvs> { encoding = {OT1,U},
6488 <zpeu> family = zpeu }
6489 <euroitc> family = {euroitc,euroitcs} }
6490 <mvs> family = mvs }
6491 {
6492 <zpeu> E = {50, }
6493 <euroitc> E = {100,50}
6494 <mvs> 164 = {50,50}, % \EUR
6495 <mvs> 068 = {50,-100} % \EURdig
6496 }
6497
6498 <*zpeu|euroitc>
6499 \SetProtrusion
6500 { encoding = U,
6501 <zpeu> family = zpeu,
6502 <euroitc> family = {euroitc,euroitcs},
6503 shape = it* }
6504 {
6505 <zpeu> E = {100,-50}
6506 <euroitc> E = {100,}
6507 }
6508
6509 </zpeu|euroitc>
6510 <*zpeu>
6511 \SetProtrusion
6512 { encoding = U,
6513 family = {zpeus,eurosans} }
6514 {
6515 E = {100,50}
6516 }
6517
6518 \SetProtrusion
6519 { encoding = U,
6520 family = {zpeus,eurosans},
6521 shape = it* }
6522 {
6523 E = {200, }

```

¹⁴ Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

Figure 1: Example for interword spacing (from Siemoneit 1989). The numbers indicate the preference/order when the interword space needs to be shrunk.



```
6524   }
6525
6526 </zpeu>
6527 </cfg-e>
```

15.9 Interword spacing

Default unit is space.

```
6528 <*m-t>
6529 %%% -----
6530 %%% INTERWORD SPACING
6531
6532 \SetExtraSpacing
6533   [ name = default ]
6534   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6535   {
```

These settings are only a first approximation. The following reasoning is from a mail from *Ulrich Dirr*, who also provided the sample in figure 1. 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
- 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)
- [before or] after lowercase characters with ascenders

```
6536   { , } = { , -500, 500 },
6537   r = { , -300, 300 },
6538   b = { , -200, 200 },
6539   d = { , -200, 200 },
6540   f = { , -200, 200 },
6541   h = { , -200, 200 },
6542   k = { , -200, 200 },
6543   l = { , -200, 200 },
6544   t = { , -200, 200 },
```

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

```

6545         c = { , -100, 100 },
6546         p = { , -100, 100 },
6547         v = { , -100, 100 },
6548         w = { , -100, 100 },
6549         z = { , -100, 100 },
6550         x = { , -100, 100 },
6551         y = { , -100, 100 },

```

- [before or] after lowercase characters with x-height plus descender without additional optical space

```

6552         i = { , 50, -50 },
6553         m = { , 50, -50 },
6554         n = { , 50, -50 },
6555         u = { , 50, -50 },

```

- after colon and semicolon

```

6556         : = { , 200, -200 },
6557         ; = { , 200, -200 },

```

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

```

6558         . = { , 250, -250 },
6559         ! = { , 250, -250 },
6560         ? = { , 250, -250 }

```

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

```

6561     }
6562

```

Questions are:

- Is the result really better?
- Is it overdone? (Try with a factor < 1000.)
- Should the first parameter also be used? (Probably.)
- What about quotation marks, parentheses etc.?

15.9.1 Nonfrenchspacing

The following settings simulate `\nonfrenchspacing` (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the `TEXbook`:

‘If the space factor f is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [...] Then the stretch component is multiplied by $f/1000$, while the shrink component is multiplied by $1000/f$.’

The ‘extra space’ (`\fontdimen 7`) for Computer Modern Roman is a third of `\fontdimen 2`, i. e., 333.

```

6563 \SetExtraSpacing
6564 [ name = nonfrench-cmr,

```

```

6565     load      = default,
6566     context    = nonfrench ]
6567   { encoding = {OT1,T1,LY1,OT4,QX,T5},
6568     family     = cmr }
6569   {

```

latex.ltx has:

```

\def\nonfrenchspacing{
  \sfcode`\ . 3000
  \sfcode`\? 3000
  \sfcode`\! 3000

```

```

6570     . = {333,2000,-667},
6571     ? = {333,2000,-667},
6572     ! = {333,2000,-667},

```

```

\sfcodes\ : 2000

```

```

6573     : = {333,1000,-500},

```

```

\sfcodes\ ; 1500

```

```

6574     ; = {    , 500,-333},

```

```

\sfcodes\ , 1250

```

```

6575     {,}= {    , 250,-200}

```

```

}

```

```

6576   }
6577

```

fontinst, however, which is also used to create the PSNFSS font metrics, sets `\fontdimen 7` to 240 by default. Therefore, the fallback settings use this value for the first component.

```

6578 \SetExtraSpacing
6579 [ name      = nonfrench-default,
6580   load      = default,
6581   context    = nonfrench ]
6582 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6583 {
6584   . = {240,2000,-667},
6585   ? = {240,2000,-667},
6586   ! = {240,2000,-667},
6587   : = {240,1000,-500},
6588   ; = {    , 500,-333},
6589   {,}= {    , 250,-200}
6590 }
6591

```

15.10 Additional kerning

Default unit is 1em.

```

6592 %%% -----
6593 %%% ADDITIONAL KERNING
6594

```

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

```
6595 \SetExtraKerning
6596   [ name = empty ]
6597   { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
6598   { }
6599
```

15.10.1 French

The ratio of `\fontdimen 2` to `\fontdimen 6` varies for different fonts, so that either the kerning of the colon (which should be a space, i. e., `\fontdimen 2`) or that of the other punctuation characters (T_EX's `\thinspace`, i. e., one sixth of `\fontdimen 6`) may be inaccurate, depending on which unit we choose (space or 1em). For Times, for example, a thin space would be 665. I don't know whether French typography really wants a thin space, or rather (as it happens to turn out with CMR) half a space. (Wikipedia¹⁵ claims it should be a quarter of an em, which seems too much to me; then again, it also says that this *was* a thin space in French typography.)

```
6600 \SetExtraKerning
6601   [ name      = french-default,
6602     context    = french,
6603     unit       = space ]
6604   { encoding = {OT1,T1,LY1} }
6605   {
6606     : = {1000,}, % = \fontdimen2
6607     ; = {500, }, % ~ \thinspace
6608     ! = {500, },
6609     ? = {500, }
6610   }
6611
```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfT_EX.

```
6612 \SetExtraKerning
6613   [ name      = french-guillemets,
6614     context    = french-guillemets,
6615     load       = french-default,
6616     unit       = space ]
6617   { encoding = {T1,LY1} }
6618   {
6619     \guillemotleft = { ,800}, % = 0.8\fontdimen2
6620     \guillemotright = {800, }
6621   }
6622
6623 \SetExtraKerning
6624   [ name      = french-guillemets-OT1,
6625     context    = french-guillemets,
6626     load       = french-default,
6627     unit       = space ]
6628   { encoding = OT1 }
6629   { }
6630
```

15.10.2 Turkish

```
6631 \SetExtraKerning
6632   [ name      = turkish,
```

15 http://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.


```

6633     context = turkish ]
6634     { encoding = {OT1,T1,LY1} }
6635     {
6636         : = {167, }, % = \thinspace
6637         ! = {167, },
6638         {=} = {167, }
6639     }
6640
6641 </m-t>
6642 </config>

```

16 Auxiliary file for micro fine tuning

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

```

6643 <!--test>
6644 \documentclass{article}
6645
6646 %% Here you can specify the font you want to test, using
6647 %% the commands \fontfamily, \fontseries and \fontshape.
6648 %% Make sure to end all lines with a comment character!
6649 \newcommand*\TestFont{%
6650     \fontfamily{ppl}%
6651     %% \fontseries{b}%
6652     %% \fontshape{it}% sc, sl
6653 }
6654
6655 \usepackage{ifthen}
6656 \usepackage[T1]{fontenc}
6657 \usepackage[latin1]{inputenc}
6658 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
6659
6660 \pagestyle{empty}
6661 \setlength{\parindent}{0pt}
6662 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
6663 \newcommand*\testprotrusion[2][]{%
6664     \ifthenelse{\equal{#1}{r}}{\{#2}%
6665         lorem ipsum dolor sit amet,
6666         \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
6667         \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6668         you know the rest%
6669     \ifthenelse{\equal{#1}{l}}{\{#2}%
6670     \linebreak
6671     {\fontencoding{\encodingdefault}%
6672     \fontseries{\seriesdefault}%
6673     \fontshape{\shapedefault}%
6674     \selectfont
6675     Here is the beginning of a line, \dotfill and here is its end}\linebreak
6676 }
6677 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
6678 \def\stripprefix#1>{}
6679 \newcount\charcount
6680 \begin{document}
6681
6682 \microtypesetup{expansion=false}
6683
6684 {\centering The font in this document is called by:\\
6685 \texttt{\showTestFont}\par}\bigskip
6686
6687 \TestFont\selectfont
6688 This line intentionally left empty\linebreak
6689 %% A -- Z
6690 \charcount=65

```

```

6691 \loop
6692   \testprotrusion{\char\charcount}
6693   \advance\charcount 1
6694   \ifnum\charcount < 91 \repeat
6695 %% a -- z
6696 \charcount=97
6697 \loop
6698   \testprotrusion{\char\charcount}
6699   \advance\charcount 1
6700   \ifnum\charcount < 123 \repeat
6701 %% 0 -- 9
6702 \charcount=48
6703 \loop
6704   \testprotrusion{\char\charcount}
6705   \advance\charcount 1
6706   \ifnum\charcount < 58 \repeat
6707 %%
6708   \testprotrusion[r]{,}
6709   \testprotrusion[r]{.}
6710   \testprotrusion[r]{;}
6711   \testprotrusion[r]{:}
6712   \testprotrusion[r]{?}
6713   \testprotrusion[r]{!}
6714   \testprotrusion[l]{\textexclamdown}
6715   \testprotrusion[l]{\textquestiondown}
6716   \testprotrusion[r]{}}
6717   \testprotrusion[l]{(}
6718   \testprotrusion{/}
6719   \testprotrusion{\char`\}
6720   \testprotrusion{-}
6721   \testprotrusion{\textendash}
6722   \testprotrusion{\textemdash}
6723   \testprotrusion{\textquoteleft}
6724   \testprotrusion{\textquoteright}
6725   \testprotrusion{\textquotedblleft}
6726   \testprotrusion{\textquotedblright}
6727   \testprotrusion{\quotesinglbase}
6728   \testprotrusion{\quotedblbase}
6729   \testprotrusion{\guilsinglleft}
6730   \testprotrusion{\guilsinglright}
6731   \testprotrusion{\guillemotleft}
6732   \testprotrusion{\guillemotright}
6733
6734 \newpage
6735 The following displays the current font stretched by 5%,
6736 normal, and shrunk by 5%:
6737
6738 \bigskip
6739 \newlength{\MTln}
6740 \newcommand*\teststring
6741   {ABCDEFGH IJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
6742 \settowidth{\MTln}{\teststring}
6743 \microtypesetup{expansion=true}
6744
6745 \parbox{1.05\MTln}{\teststring\linebreak\}
6746   \teststring\par\bigskip
6747 \parbox{0.95\MTln}{\teststring}
6748
6749 \end{document}
6750 </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 <i>Harald Harders</i>)	78	<code>\MT@get@basefamily</code> : only remove suffix if it is ‘x’ or ‘j’	79
issue an error instead of a warning, when pdfTeX version is too old for autoexpand	121	<code>\MT@get@listname@</code> : don’t check for empty attributes list	80
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i>)	129	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i>)	42
Protrusion: add factors for some more characters settings for Adobe Minion (contributed by <i>Harald Harders</i>)	134	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i>)	84
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance	135	<code>\MT@pdftex@no</code> : fix: version check (reported by <i>Harald Harders</i>)	38
<code>\MT@declare@sets</code> : remove spaces around set name	93	<code>\MT@permute</code> : don’t use sets for empty encoding ..	107
<code>\MT@DeclareSet</code> : remove spaces around first argument	92	<code>\MT@split@codes</code> : fix: allow zero and negative values ..	58
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded ..	78	<code>\MT@use@set</code> : remove spaces around set name	97
		<code>\UseMicrotypeSet</code> : remove spaces around first argument	96

Version 1.2 (2004/10/03)

Font sets: declare cmr as an alias of cmr	127	changed	93
new: allmath and basicmath	126	<code>\MT@get@inh@list</code> : fix: set inheritance list \globally to \empty	82
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding	155	<code>\MT@get@listname@</code> : alternatively check for alias font name	80
add settings for Computer Modern Roman math symbols	160	<code>\MT@get@size</code> : additional magic to catch some errors hijack \set@fontsize instead of \setfontsize ..	95
<code>\MT@encoding@check</code> : check whether only one encoding specified	106	<code>\MT@loop</code> : fix: new macro, used instead of \loop ..	46
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement	55	<code>\MT@maybe@do</code> : also check for alias font name	55
<code>\MT@get@basefamily</code> : also remove ‘w’ (swash capitals)	79	<code>\MT@permute@@@@@</code> : more sanity checks for \SetProtrusion and \SetExpansion	108
<code>\MT@get@highlevel</code> : check whether defaults have		<code>\MT@setupfont</code> : also search for alias font file	52
		fix: call \@@enc@update if necessary	52

Version 1.3 (2004/10/27)

General: fix: specifying load option does no longer require to give a name, too	103	<code>\MT@fix@catcode</code> : check some category codes (compatibility with german)	34
Font sets: declare aer, zer and hfor as aliases of cmr	127	<code>\MT@load@list</code> : check whether list exists	78

Version 1.4 (2004/11/12)

General: check for pdfcprot	50	(OT1, T1, lmr)	140
don’t use scratch registers in global definitions ..	82	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options	116
use \pickup@font instead of \define@newfont as the hook for \MT@setupfont	87	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too	100
use one instead of five counters	47		
Protrusion: tweak quote characters for cmr variants			

Version 1.4a (2004/11/17)

General: new option: final	113	when reading files (reported by <i>Michael Hoppe</i>)	79
\MT@cfg@catcodes: fix: reset some more catcodes			

Version 1.4b (2004/11/26)

General: fix: set catcodes before reading global configuration file (reported by <i>Christoph Bier</i>)	115	name if encoding failed	56
new message if \pdfoutput is changed	119	\MT@get@basefamily: fix: failed for font names of the form abczz (reported by <i>Georg Verwey</i>)	79
optimisation: use less \expandafters and \csnames	41	\MT@get@slot: don't define \MT@char globally (save stack problem)	82
Protrusion: harmonise dashes in upshape and italic (cmr, pad, ppl)	134	\MT@ifdimen: don't set \MT@count globally (save stack problem)	43
slanted like italics	143	\MT@use@set: don't use undeclared font sets	97
\MT@checklist@family: fix: don't try alias family			

Version 1.5 (2004/12/15)

General: defaults: step: 4 (suggested by <i>Hàn Thế Thành</i>)	114	\MT@cfg@catcodes: reset catcode of '=' (compatibility with Turkish babel)	79
defaults: calculate step as min(stretch,shrink)/5	120	\MT@fix@catcode: reset catcode of '~ (compatibility with chemsym)	34
defaults: turn off expansion for DVI output	119	\MT@get@highlevel: don't test defaults if called after begin document	93
disable automatic expansion for DVI output	121	\MT@scale@factor: warning for factors outside limits	60
new option: selected, by default false (suggested by <i>Hàn Thế Thành</i>)	112	\MT@scale@to@em: don't use \lpcode and \rprcode for the calculation	59
Documentation: add 'Short history'	29	\MT@set@ex@codes: allow non-selected font expansion	64
add note about DVIoutput option	9	\MT@set@pr@codes: adjust protrusion factors before setting the inheriting characters	57
Inheritance: remove \ss from T1 list, add \DJ	129		
Protrusion: settings for Bitstream Charter	135		
\DeclareMicrotypeAlias: remove spaces around arguments	98		

Version 1.6 (2005/01/24)

General: defaults: turn off expansion for old pdfTeX versions	115	improve settings for numbers (pointed out by <i>Peter Muthesius</i>)	137
disable automatic expansion for old pdfTeX versions	121	tune CMR math letters (OML encoding)	160
load a font if none is selected	52	\MT@get@charwd: use e-TeX's \fontcharwd, if available	59
new option: factor, by default 1000	114	\MT@get@inh@list: correct message if selected is false	81
restructure dtx file	126	\MT@set@ex@codes: introduce factor option	64
test whether \pickup@font has changed	89	\MT@set@pr@codes: introduce factor option	57
test whether numeric options receive a number	114	\MT@use@set: retain current set if new set is undeclared	97
use e-TeX's \ifcsname and \ifdefined if defined	42	\MT@vinfo: new macro instead of \ifMT@verbose	35
Protrusion: add italic uppercase Greek letters	143		

Version 1.6a (2005/02/02)

Documentation: add table of fonts with tailored protrusion settings	21	reported by <i>Bernard Gaulle</i>)	82
\MT@get@slot: completely redone, hopefully more robust (compatible with frenchpro; problem		\MT@pdf@tex@no: new macro	37
		\MT@reset@ef@codes: only reset \efcodes for older pdfTeX versions	65

Version 1.7 (2005/03/23)

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i>)	94	fix: remove space after autoexpand	105
disallow automatic expansion if pdfTeX too old	105	modify \showhyphens	122
		new value for verbose option: errors	114

shorter command names	47	test whether <code>\(encoding)\(...</code> is defined	83
warning when running in draft mode	119	<code>\MT@if@list@exists</code> : don't define <code>\MT@#1@c@name</code>	
Documentation: add hint about compatibility	27	globally, here and elsewhere	81
remove table of match order	12	<code>\MT@if@dimen</code> : comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i>)	43
Protrusion: fix: remove <code>\</code> from OT1, add <code>\textbackslash</code> to T1 encoding	138	<code>\MT@increment</code> : use e-TeX's <code>\numexpr</code> if available	47
<code>\DeclareMicrotypeAlias</code> : may also be used inside configuration files	98	<code>\MT@is@composite</code> : new macro: construct command for composite character; no uncontrolled expansion	85
<code>\LoadMicrotypeFile</code> : new command (suggested by <i>Andreas Böhmann</i>)	98	<code>\MT@scale</code> : new macro: use e-TeX's <code>\numexpr</code> if available	47
<code>\Microtype@Hook</code> : new command for font package authors	116	<code>\MT@set@ex@codes</code> : two versions of this macro	64
<code>\microtypesetup</code> : fix: warning also when setting to (no)compatibility	116	<code>\MT@split@name</code> : don't define <code>\MT@encoding &c.</code> globally	54
<code>\MT@begin@catcodes</code> : also use inside configuration commands	79	<code>\MT@test@ast</code> : make it simpler	93
<code>\MT@cfg@catcodes</code> : reset catcode of <code>'</code> (compatibility with french* packages)	79	<code>\MT@try@order</code> : always check for size, too (suggested by <i>Andreas Böhmann</i>)	80
<code>\MT@get@listname@</code> : use <code>\etfor</code> (<i>Andreas Böhmann</i> 's idea)	80	fix: also check for <code>//(series)/(shape)//</code> (reported by <i>Andreas Böhmann</i>)	80
<code>\MT@get@slot</code> : remove backslash hack	82	<code>\MT@warn@code@too@large</code> : new macro: type out maximum protrusion factor	60
test for <code>\chardef</code> commands	83	<code>\MT@warn@err</code> : new macro: for verbose=errors	35

Version 1.8 (2005/06/23)

General: <code>\SetProtrusion</code> : new key: unit	104	<code>\MT@find@file</code> : no longer wrap names in commands	78
if font substitution has occurred, set up the substitute font, not the selected one	87	<code>\MT@get@charwd</code> : warning for missing (resp. zero-width) characters	59
new option: config to load a different main configuration file	115	<code>\MT@get@font@dimen@six</code> : new macro: test whether <code>\fontdimen 6</code> is defined	57
new option: unit, by default character	114	<code>\MT@get@listname@</code> : made recursive	80
Documentation: add example for factor option	13	<code>\MT@get@slot</code> : fix: expand active characters	82
add example of how to get rid of a widow (suggested by <i>Adam Kucharczyk</i>)	15	test whether <code>\(encoding)\(...</code> is defined made more robust	83
add hint about error messages	27	<code>\MT@get@unit</code> : new macro: get unit for codes	62
Font sets: add U encoding to allmath	126	<code>\MT@in@list</code> : made recursive	46
declare <code>pxr</code> and <code>txr</code> as aliases of <code>ppl</code> resp. <code>ptm</code>	127	<code>\MT@is@active</code> : new macro: translate inputenc-defined characters	84
Inheritance: remove <code>\DJ</code> from T1 list (it's the same as <code>\DH</code>)	129	<code>\MT@is@letter</code> : warning for non-ASCII characters	83
Protrusion: add LY1 characters for Times	142	<code>\MT@led@kern</code> : character protrusion with <code>ledmac</code>	49
settings for AMS math fonts	164	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\@for</code>	45
verified settings for slanted Computer Modern Roman	149	<code>\MT@map@tlist@n</code> : new macro: used instead of <code>\@tfor</code>	45
<code>\add@accent</code> : fix: disable micro-typographic setup inside <code>\add@accent</code> (reported by <i>Stephan Hennig</i>)	89	<code>\MT@old@cmd</code> : renamed commands from <code>\..MicroType..</code> to <code>\..Microtype..</code>	35
<code>\DeclareMicrotypeAlias</code> : warning when overriding an alias font	98	<code>\MT@pdftex@no</code> : case 5: pdfTeX 1.30	37
<code>\DeclareMicrotypeSetDefault</code> : new command: set default font set	97	<code>\MT@permute@@@@@</code> : add ranges to the beginning of the lists	109
<code>\MT@cfg@catcodes</code> : reset catcodes of the remaining ASCII characters	79	<code>\MT@scale</code> : fix: remove spaces in e-TeX variant (reported by <i>Mark Rossi</i>)	47
<code>\MT@check@rlist</code> : made recursive	109	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>hyperref</code> is loaded	50
<code>\MT@curr@list@name</code> : new macro: current list type and name	86	restore csquotes's active characters	50
<code>\MT@declare@sets</code> : warning when redefining a set	93	restore percent character if Spanish <code>babel</code> is loaded	50
<code>\MT@define@set@key@</code> : use comma lists instead of token lists	93	<code>\MT@split@codes</code> : get character width once only	58
		<code>\MT@use@set</code> : fix: remove braces in first line	97
		<code>\MT@xadd</code> : simplified	44

Version 1.9 (2005/10/28)

General: <code>\DeclareMicrotypeSet</code> : new key: font	95	<code>\SetProtrusion</code> : value 'relative' renamed to	
--	----	---	--

‘character’ for key unit	104	settings for T5 encoded Computer Modern Roman	134
allow context-specific font setup	87	\DisableLigatures: new command: disable ligatures	
disable expansion if both step and shrink are zero	121	(requires pdf \TeX 1.30)	99
disable microtype setup inside hyperref’s		\microtypecontext: new command: change setup	
\pdfstringdef (reported by <i>Hàn Thế Thành</i>)	51	context in the document	91
fix: use true as the default value	111	\MT@checklist@family: fix: add two missing	
option unit: rename value relative to character	114	\expandafers	56
warning if user requested zero step	120	\MT@detokenize@c: fix the \mathcal{E} - \TeX version	42
Documentation: add hint about verbatim environment	26	\MT@exp@two@n: new macros: less \expandafers	42
add remark about Type 1 fonts required for automatic font expansion	8	\MT@get@opt: new key ‘preset’ to set all characters to the specified value before loading the lists	62
Font sets: add OT4 encoding to text sets	126	\MT@is@active: redone: use \set@display@protect	84
add T5 encoding to text sets	126	\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning	83
declare qpl and qtm (qfonts, \TeX Gyre) as aliases of ppl resp. ptm	127	\MT@maybe@do: redone	55
Inheritance: add list for OT4	131	\MT@pdf@tex@no: compatibility with \TeX Live hack (reported by <i>Herbert Voß</i>)	38
add list for T5 (requested by <i>Hàn Thế Thành</i>)	132	\MT@rem@from@clist: new macro: remove an item from a comma list	46
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR	139	\MT@scale@factor: generalised	60
settings for OT4 encoding (Computer Modern Roman, Palatino, Times)	134	\MT@toks: use instead of \toks@	39
		\SetProtrusion: (et al.) new key: font	99

Version 1.9a (2005/12/05)

General: ‘(file name)/(line number)’ as default list name	102	diately (requested by <i>Georg Verwey</i>)	93
new option: deferfontsetup, by default true	113	\MT@get@highlevel: no longer check whether defaults have changed	93
remove superfluous test whether \pickup@font has changed	89	\MT@ifdefined@c@T: new macros: true case only	42
Documentation: add explanation for error message in DVI mode	27	\MT@ifint: use \pdfmatch if available	43
add explanation for error message with non-Type 1 fonts	27	\MT@ifstreq: use \pdfstrcmp if available	44
Font sets: declare mdbch (mathdesign) as an alias of Charter	128	\MT@in@clist: fix	45
Protrusion: fix: remove ‘_’ from OT1 encoding	139	\MT@info@missing@char: info instead of warning (after <i>Michael Hoppe</i> reported that the ‘fl’ ligature is missing in Palatino SC)	60
settings for T5 encoded Charter	134	\MT@is@feature: new macro: check for pdf \TeX feature	48
\microtypesetup: inside the preamble, accepts all package options	116	\MT@map@clist@n: following \LaTeX 3	45
\MT@check@font@cx: optimise context-sensitive setup	90	\MT@permute@: don’t define permutations for unused encodings	108
\MT@define@set@key@: don’t expand variables immediately		\MT@rem@from@clist: fix	46
		\MT@setup@: defer setup until the end of the preamble	48

Version 1.9b (2006/01/20)

General: compatibility with listings: sanitise more catcodes (reported by <i>Holger Uhr</i>)	51	add samples of micro-typographic features	4
compatibility with the extendedchar option of the listings package	51	\MT@features: use throughout the package to adjust to beta-ness	47
Documentation: activate expansion in the distributed PDF	1	\MT@ifdimen: use \pdfmatch if available	43
		\MT@warn@code@too@large: fix calculation with present factor	60

Version 1.9c (2006/02/02)

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verwey</i>)	22	\MT@define@code@key@font: fix: context was ignored	102
Protrusion: settings for URW Garamond	135	\MT@define@code@key@size: fix: embrace \MT@temp@size in \csname (bug introduced in v1.9b)	102

Version 1.9d (2006/05/05)

Font sets: md* instead of m series in basic sets	126	\MT@get@font@dimen: warning for zero fontdimen . .	60
add QX encoding to text sets	126	\MT@get@opt: optimise: don't reset when preset op-	
Inheritance: add list for QX encoding (contributed by		tion is set	62
Maciej Eder)	131	set list name before presetting	62
Protrusion: settings for QX encoding (contributed by		\MT@is@active: support for Unicode (inputenc/utf8) .	84
Maciej Eder)	140	\MT@setupfont@hook: restore % and # when tex4ht	
settings for Euro symbols (Adobe, ITC, marvosym)	172	is loaded (reported by <i>Peter Dybala</i>)	50
tweak AMS settings	164	\SetProtrusion: (et al.) optimise: unify keys for	
\DeclareCharacterInheritance: fix: empty context	105	mandatory argument	99
\MT@detokenize@n: new macro: use \detokenize if		(et al.) split keys of optional and mandatory argu-	
available	42	ment	99
\MT@get@ex@opt: fix: evaluate preset	65		

Version 1.9e (2006/07/28)

General: fix: default value for activate: true . . .	112	settings for Euler Roman font	168
Documentation: add hint about unknown encodings	26	\DeclareCharacterInheritance: new key 'inputenc'	
include LPPL	193	to set the input encoding	105
Font sets: declare zeur and zeus (eulervm) as aliases		\MT@rem@from@cllist: model after \@removeelement	46
of eur resp. eus (euler)	128	\MT@setup@: empty \MT@setup@ after use (compatibil-	
Inheritance: adapt to marvosym's changed encoding	133	ity with the combine class)	48
Protrusion: complete settings for Euler Fraktur and		\pickup@font: no tracing with trace package	89
Script fonts	171	\SetExpansion: new key: inputenc	100
fix: forgotten comma in mt-mvs.cfg; adapt to		\SetProtrusion: (et al.) new key: inputenc	99
marvosym's changed encoding	172		

Version 1.9f (2006/09/09)

Protrusion: fix: euler-vm did not load euler settings	168	\MT@reset@context: only reset context if it has actu-	
\MT@curr@list@name: fix: \MessageBreak must not be		ally been changed	91
expanded	86	\MT@set@inh@list: fix: forgotten comma in the fea-	
\MT@gdef@n: new macros: global variants	41	tures list	106
\MT@get@inh@list: fix: input encoding must be set		\MT@set@named@keys: new macro: set name first, sim-	
after the inheritance list has been parsed	82	plify parsing of optional argument	101
\MT@glet: new macro	41	\SetProtrusion: (et al.) set catcodes before parsing	
		optional argument	99

Version 2.0 (2007/01/14)

General: (beta:1) new option: babel, by default false		Documentation: add hint about how to increase	
(language-dependent setup suggested by <i>Ulrich</i>		font_max and font_mem_size	27
<i>Dirr</i>)	112	add hint about warning when tracking and expan-	
(beta:1) new option: letterspace, by default 100	114	sion is applied to a font	28
(beta:3) compatibility with soul: register \textls		add remark about 'draft' option disabling	
and \lsstyle	51	microtype (noted by <i>Michalis Miatidis</i>)	9
(beta:8) option 'babel': fix: switch off French		qualify hint about web documents with regard to	
babel's shorthands properly (reported by <i>Daniel</i>		older pdfTeX versions	25
<i>Flipo</i>)	125	qualify hints about expansion error messages with	
(beta:8) option 'babel': switch off Turkish babel's		regard to older pdfTeX versions	27
shorthands	125	Font sets: add footnotesize and scriptsize sets	126
compatibility with listings: set catcode of back-		add smallcaps set	126
slash to zero (reported by <i>Steven Bath</i>)	51	\DeclareMicrotypeBabelHook: (beta:1) new com-	
maybe disable \MT@noligatures after the preamble	124	mand: interaction with babel	99
new package letterspace: a stripped-down ver-		\lsstyle: (beta:1) new command: letterspacing . .	72
sion, containing the letterspacing commands		(beta:8) fix: font switches don't pose a problem	
only	1	anymore	72
option 'unit', \SetProtrusion: deprecate value		(beta:8) fix: letterspacing commands may be nes-	
'relative' completely	104	ted	72

totally redone, using the new <code>\letterspacefont</code>	72	<code>\SetExtraKerning</code> : (beta:1) new command: additional kerning	101
<code>\MT@declare@sets</code> : fix: empty size list when redefining set	93	<code>\SetExtraSpacing</code> : (beta:1) new command: adjustment of interword spacing	101
<code>\MT@is@symbol</code> : made even more robust	85	<code>\SetTracking</code> : new command: tracking	100
<code>\MT@load@inputenc</code> : sanitise catcodes before loading input encoding (problem with listings)	63	<code>\textls</code> : (beta:1) new command: letterspacing	76
<code>\MT@pdfTeX@no</code> : (beta:1) case 6: pdfTeX 1.40	37	starred version: remove spaces around text	76
<code>\MT@split@name</code> : adjust to possible letterspacing	54	<code>\tracingmicrotypeinpdf</code> : new debug method: mark all fonts with PDF annotations	36

Version 2.1 (2007/01/21)

General: compatibility with CJK: also check for its definition	88	<code>\lslig</code> : new command: protect ligatures in letterspaced text	72
compatibility with pinyin: disable microtype in <code>\py@macron</code> (reported by <i>Sven Naumann</i>)	52	<code>\MT@get@ls@basefont</code> : redone: use <code>\pdfmatch</code> to make it bullet-proof	73
fix: letterspace package forgot to load <code>keyval</code>	39	<code>\textls</code> : fix: use <code>\hmode@bgroup</code>	76

Version 2.2 (2007/07/14)

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i>)	48	fix: ϵ -TeX version shouldn't use <code>\x</code> and <code>\y</code> (found by <i>Wiebke Petersen</i>)	44
new option: <code>copyfonts</code>	113	<code>\MT@is@composite</code> : more robust: expand exactly once	85
simplify key declarations	103	<code>\MT@is@symbol</code> : expand once more (for frenchpro)	85
warning if stretch or shrink aren't multiples of step	122	<code>\MT@lsfont</code> : use <code>\font@name</code> , not <code>\MT@font</code>	70
Documentation: add hint about error message with pdfTeX 1.40	27	<code>\MT@maybe@etex</code> : use catcode trickery	40
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i>)	26	<code>\MT@pdfTeX@no</code> : case 7: pdfTeX 1.40.4	37
add overview	6	<code>\MT@preset@aux@space</code> : generalised	63
logo transparency and amusement	1	<code>\MT@requires@luatex</code> : (basic) support for LuaTeX	40
Font sets: declare FPL Neu as an alias of Palatino	127	<code>\MT@set@all@pr</code> : (et al.) allow empty values	58
declare <code>chr</code> (<code>chmath</code>) as an alias of Charter (reported by <i>Geoff Vallis</i>)	128	<code>\MT@set@inputenc</code> : only load <code>inputenc</code> files if necessary	62
default set for tracking: <code>smallcaps</code>	127	<code>\MT@set@tr@codes</code> : disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	70
Inheritance: remove <code>'-'</code> \rightarrow <code>'127'</code>	130	possibility to customise interword spacing	70
Protrusion: settings for Bitstream Letter Gothic	135	<code>\MT@setupfont</code> : don't call <code>\@enc@update</code> anymore	53
Spacing: add sample	173	only add features that are available with the respective pdfTeX	53
Tracking: add ligatures that are to be disabled	133	<code>\MT@setupfont@hook</code> : restore percent character if Galician babel is loaded	50
<code>\DeclareMicrotypeVariants</code> : new command	98	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letterspaced fonts	58
<code>\DisableLigatures</code> : new optional argument: disable selected ligatures only	99	<code>\MT@tracking</code> : remember fonts that shouldn't be letterspaced	69
<code>\lslig</code> : always defined	72	<code>\MT@tracking@</code> : fix: tracking couldn't be re-enabled	69
<code>\MT@checklist@font</code> : fix: construct font name from characteristics	57	<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	123
<code>\MT@copy@font</code> : optionally work on copies of fonts	53	<code>\MT@with@babel@and@T</code> : also inspect class options	49
<code>\MT@get@basefamily</code> : redone, working on font names and suffixes of arbitrary length	79	<code>\pickup@font</code> : letterspace: setup inside group	88
<code>\MT@get@charwd</code> : subtract letterspacing amount from width	59	<code>\SetTracking</code> : new key 'no ligatures' to disable ligatures of letterspaced fonts	100
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	73	new keys 'spacing' and 'outer spacing' to adjust interword spacing (suggested by <i>Steven E. Harris</i>)	100
<code>\MT@ifdimen</code> : employ LuaTeX features if available	43	third argument may be empty	100
<code>\MT@ifint</code> : employ LuaTeX features if available	43	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code>	91
<code>\MT@ifstreq</code> : employ LuaTeX features if available	44		

Version 2.3 (2007/12/23)

General: disable \microtypecontext in hyperref's \pdfstringdef	51	\MT@in@clist: fix: don't use \x (reported by <i>Peter Meier</i>)	45
enable protrusion when tracking is enabled ...	122	\MT@is@active: support for extended Unicode (inputenc/utf8x resp. ucs) – experimental ...	84
fix: really switch off Turkish shorthands	125	\MT@noligatures: fix: set evaluation didn't work (bug introduced in 2.2)	77
new value for verbose option: silent (suggested by <i>Karl Berry</i>)	114	\MT@plain: letterspace: support for eplain/miniltx	37
turned some warnings into errors	114	\MT@set@curr@os: adjusting spaces made more reli- able	71
Documentation: add kerning sample	20	\MT@set@tr@codes: also adjust tracking if protrusion is not enabled, and even for letterspace (reported by <i>Stephan Hennig</i>)	70
add letterspacing illustration	17	possibility to customise outer kerning (suggested by <i>Stephan Hennig</i>)	70
\do@subst@correction: remember substitute font for all times (reported by <i>Stephan Hennig</i>)	89	\MT@SetTracking: sanity check for value	100
\lslig: redone: extract outer kerns from current letterspacing amount	72	\MT@tr@outer@l: only change pre outer space if it contains shrink	74
\microtypecontext: made robust (reported by <i>Ste- phan Hennig</i>)	91		
\MT@begin@catcodes: fix: don't disable \KV@sp@def	79		
\MT@define@set@key@font: font: single asterisk means normal font	95		

Version 2.3a (2008/02/29)

General: error messages if pdfTeX is too old for exten- sions	123	\MT@fix@catcode: fix catcodes earlier, and also for the letterspace package	34
fix test for soul under plain TeX	51	\MT@getkey: fix: key=val in class options list ...	119
Documentation: add hint about babel having to be loaded first	27	\MT@maybe@etex: removed	40
add table of available and enabled features	7	\MT@set@codes: generalised	58
mention soulutf8	29	\MT@setupfont@hook: restore percent character if Mexican babel is loaded	50
Protrusion: adjust LMR quotation marks again ...	140		

Version 2.3b (2008/06/04)

General: compatibility with CJKutf8: also check for its definition	88	known slots	77
\MT@exp@gcs: new macro: reduce save stack size ..	41	\MT@requires@latex: new macro	37
\MT@font@copy: enable font copies also with protru- sion contexts (reported by <i>Nathan Rosenblum</i>)	53	\MT@set@tr@codes: fix: protrusion adjustment only for new fonts (reported by <i>Wolfram Schaalo</i>) ..	70
\MT@get@size: grouping	95	\MT@tr@outer@l: fix: only in horizontal mode ...	74
\MT@noligatures@: fix: warning messages for un- known slots	77	make \spaceskip-aware (ragged2e)	74
		\MT@tr@outer@r@: additional test for horizontal mode	75

B Index

Links (in blue) refer to the page where the corresponding entry is described (bold face) resp. occurs. Plain numbers (in black) refer to the code line where the corresponding entry is defined (underlined>) resp. used.

Options:		protrusion	6
DVIoutput	9	selected	8
activate	6	shrink	8
auto	8	spacing	7
babel	9	step	8
config	9	stretch	8
draft	9	tracking	7
expansion	6	unit	8
factor	7	verbose	9
final	9		
kerning	7	User Commands:	
letterspace	9	\DeclareCharacterInheritance	20

- `\DeclareMicrotypeAlias` 22
 - `\DeclareMicrotypeBabelHook` 23
 - `\DeclareMicrotypeSet` 10
 - `\DeclareMicrotypeSet*` 10
 - `\DeclareMicrotypeSetDefault` 11
 - `\DeclareMicrotypeVariants` 21
 - `\DeclareMicrotypeVariants*` 21
 - `\DisableLigatures` 25
 - `\LoadMicrotypeFile` 22
 - `\lslig` 24
 - `\lstyle` 24
 - `\microtypecontext` 22
 - `\microtypesetup` 10
 - `\SetExpansion` 14
 - `\SetExtraKerning` 19
 - `\SetExtraSpacing` 18
 - `\SetProtrusion` 13
 - `\SetTracking` 16
 - `\textls` 24
 - `\textls*` 24
 - `\textmicrotypecontext` 22
 - `\UseMicrotypeSet` 11
- A**
- `a0poster` (package) 95
 - `activate` (option) 6, 112, 183
 - `\add@accent` 2322
 - `ae` (package) 21, 127
 - `amssymb` (package) 164
 - `article` (package) 23
 - `auto` (option) 8, 112
- B**
- `babel` (option) 9, 23, 27, 30, 112, 183
 - `babel` (package) 3, 5, 17, 23, 24, 27, 50, 79, 99, 124, 125, 128, 180, 181, 183–185
- C**
- `chemsym` (package) 180
 - `chmath` (package) 21, 128, 184
 - `CJK` (package) 52, 88, 184
 - `CJKutf8` (package) 88, 185
 - `cm-super` (package) 8
 - `color` (package) 9, 119
 - `combine` (package) 48, 183
 - `config` (option) 9, 21, 31, 115, 181
 - `contour` (package) 119
 - `copyfonts` (option) 53, 112, 113, 184
 - `crop` (package) 119
 - `csquotes` (package) 28, 50, 138, 181
 - `\curr@fontshape` 2269, 2271, 2273, 2280, 2320
- D**
- `\DeclareCharacterInheritance` 20, 41, 3041
 - `\DeclareMicrotypeAlias` 22, 43, 70, 2698
 - `\DeclareMicrotypeBabelHook` 23, 45, 64, 2751
 - `\DeclareMicrotypeSet` 10, 32, 57, 59, 71, 2429
 - `\DeclareMicrotypeSet*` 10, 2429
 - `\DeclareMicrotypeSetDefault` 11, 34, 61, 2657
 - `\DeclareMicrotypeVariants` 21, 42, 58, 63, 2685
 - `\DeclareMicrotypeVariants*` 21
 - `defersetsup` (option) 50, 112, 113, 116, 182
 - `\define@newfont` 2262, 2270, 2274, 2282
- `\DisableLigatures` 25, 40, 62, 2731
 - `\do@subst@correction` 2319
 - `docstrip` (package) 33
 - `draft` (option) 9, 112, 113, 183
 - `dsfont` (package) 57
 - `DVIoutput` (option) 9, 112, 180
- E**
- `eco` (package) 21, 127
 - `\efcode` 1282, 1306, 1307, 1347, 1349
 - `eplain` (package) 25, 29, 33, 37, 48, 119, 185
 - `eucal` (package) 169
 - `eufrak` (package) 171
 - `euler` (package) 86, 168, 183
 - `eulervm` (package) 21, 128, 168, 183
 - `euroitc` (package) 172
 - `europs` (package) 172
 - `eurosans` (package) 172
 - `expansion` (option) 6, 7, 11, 111, 112, 119
- F**
- `\f@family` 2276, 2277
 - `\f@size` 2269, 2271, 2273, 2280, 2320
 - `factor` (option) 7, 13, 31, 114, 180
 - `fancyvrb` (package) 27
 - `final` (option) 9, 32, 112, 113, 180
 - `fix-cm` (package) 54
 - `\font` 784, 1640
 - `\font@name` 109, 820, 822, 824, 826, 836, 837, 1530, 1535, 1569, 1576, 1586, 1589, 1628, 1640, 1652, 1654, 1659, 1661, 1664, 1693, 1748, 1841, 1843, 2262, 2270, 2279, 2282, 2303, 2305, 2307, 2321, 2333, 2353
 - `\fontcharwd` 1046
 - `\fontdimen` 975, 982, 1067, 1072, 1171, 1174, 1547, 1548, 1576, 1652, 1686, 1687, 1694, 1695, 1701, 1705, 1712, 1722, 1726, 1729, 1748, 1854, 1857
 - `fontinst` (package) 175
 - `fontinstallationguide` (package) 154
 - `french` (package) 181
 - `frenchpro` (package) 85, 180, 184
- G**
- `german` (package) 34, 179
 - `graphics` (package) 9, 119
- H**
- `hfoldsty` (package) 21, 127
 - `hyperref` (package) 9, 50, 51, 87, 119, 181, 182, 185
- I**
- `IEEEtran` (package) 87
 - `\iffontchar` 1077
 - `\ifMT@auto` 219, 3702, 3749
 - `\ifMT@babel` 219, 3874, 3921
 - `\ifMT@do` 856, 920, 1509, 1873
 - `\ifMT@document` 261, 3157
 - `\ifMT@draft` 219, 3640
 - `\ifMT@expansion` 219, 3670, 3746
 - `\ifMT@if@` 218, 687, 693, 711, 742, 2347, 3260, 3941, 3944
 - `\ifMT@inannot` 98

- `\ifMT@inlist@` 516, 561, 620,
 644, 799, 887, 904, 913, 949, 1507,
 1924, 1935, 1984, 2032, 2341, 2403, 2415, 2716
`\ifMT@kerning` 219, 3814, 3863, 3934
`\ifMT@noligatures` 219, 3890
`\ifMT@nonselected` 1260, 2045, 2053
`\ifMT@noreset` 2107, 2114
`\ifMT@opt@auto` 3267, 3705
`\ifMT@opt@DVI` 3267, 3656
`\ifMT@opt@expansion` 3267, 3651
`\ifMT@protrusion` 219, 650, 672, 3658, 3797
`\ifMT@selected` 219, 3732, 3752
`\ifMT@spacing` 219, 3803, 3857, 3873
`\ifMT@tracking` 219, 3792, 3851
`ifpdf` (package) 9, 88, 119
`\iftracingmicrotypeinpdfall` 103
`inputenc` (package) 14,
 27, 30, 51, 62, 63, 82, 84, 86, 181, 183–185
- J**
- `jurabib` (package) 87
- K**
- `kerning` (option) 7, 11, 30, 112
`keyval` (package) 39, 58, 128, 184
`\knaccode` 1462, 1463, 1473, 1476, 1482
`\knbccode` 1455, 1456, 1472, 1475, 1481
`\knbscode` 1376, 1377, 1400, 1404, 1410
- L**
- `ledmac` (package) 28, 31, 49, 87, 181
`ledpar` (package) 28
`\leftmarginkern` 658
`letterspace` (option) . 9, 18, 24, 30, 43, 69, 114, 183
`letterspace` (package) 1,
 24, 25, 29, 30, 33, 37, 49, 70, 72, 183–185
`\letterspacefont` 1535
`lineno` (package) 49
`listings` (package) 27, 51, 63, 79, 182–184
`lmodern` (package) 8, 140
`\LoadMicrotypeFile` 22, 44, 73, 2712, 3645
`\lcode` 988, 1020, 1021, 1207, 1210, 1560
`\lslig` 24, 54, 1645, 1648
`\lsstyle` 24, 51,
 734, 757, 762, 1629, 1804, 1807, 1821, 3648
- M**
- `marvosym` (package) 21, 30, 33, 133, 172, 183
`mathdesign` (package) 21, 128, 182
`memoir` (package) 23, 87
`\Microtype@Hook` 116, 3497
`\microtypecontext`
 .. 22, 47, 732, 2373, 2383, 3647, 3904, 3907
`microtypecontext` (environment) 22
`\microtypesetup` 10, 46, 3501, 3646
`miniltx` (package) 25, 29, 33, 37, 48, 185
`minimal` (package) 52
`\MT@font` 316, 802,
 818, 977, 1068, 1081, 1520, 1987, 2058, 2308
`\MT@abbr@ex` 596
`\MT@abbr@ex@c` 596
`\MT@abbr@ex@inh` 596
`\MT@abbr@kn` 596
`\MT@abbr@kn@c` 596
`\MT@abbr@kn@inh` 596
`\MT@abbr@nl` 596
`\MT@abbr@pr` 596
`\MT@abbr@pr@c` 596
`\MT@abbr@pr@inh` 596
`\MT@abbr@sp` 596
`\MT@abbr@sp@c` 596
`\MT@abbr@sp@inh` 596
`\MT@abbr@tr` 596
`\MT@abbr@tr@c` 596
`\MT@active@features`
 828, 2335, 2338, 2350, 2357, 2392, 2402,
 2738, 3659, 3747, 3794, 3805, 3816, 3896, 3897
`\MT@addto@annot` 98
`\MT@addto@setup` 635, 637, 696,
 1184, 2256, 2373, 2374, 2936, 2954, 3371,
 3502, 3639, 3790, 3850, 3889, 3895, 3920, 3953
`\MT@auto` 1269, 3701, 3715, 3725, 3729
`\MT@auto@` 1269, 1278, 1320
`\MT@autofalse` 221, 3713, 3724
`\MT@autotruer` 221, 3461
`\MT@babelfalse` 228
`\MT@babeltrue` 228
`\MT@begin@catcodes` 1925, 1926, 1966, 2720, 2733,
 2758, 2772, 2794, 2813, 2827, 3046, 3474, 3475
`\MT@cat` 1180, 2248, 2250
`\MT@cfg@catcodes` 744, 1202, 1951, 1968
`\MT@char` 1010, 1020, 1021, 1026, 1027, 1030, 1032,
 1046, 1047, 1077, 1207–1211, 1306, 1307,
 1309, 1310, 1347–1349, 1376, 1377, 1383,
 1384, 1390, 1391, 1394, 1395, 1400–1405,
 1455, 1456, 1462, 1463, 1466, 1467, 1472–
 1476, 1884, 2087, 2103, 2104, 2109, 2180,
 2183, 2185, 2211, 2214, 3106, 3107, 3112, 3113
`\MT@char@`
 2087, 2093, 2098, 2103, 2117, 2119, 2125,
 2126, 2128, 2141, 2142, 2145, 2146, 2149,
 2150, 2154, 2156, 2184, 2197, 2200, 2203, 2222
`\MT@charstring` 2100, 2183, 2188
`\MT@check@active@set`
 3489, 3665, 3767, 3796, 3808, 3820
`\MT@check@font` 798, 2332, 2394
`\MT@check@font@cx` 2336, 2394
`\MT@check@range` 3231, 3233
`\MT@check@range@` 3233, 3234
`\MT@check@rlist` 3184, 3224
`\MT@check@rlist@` 3224, 3225
`\MT@check@step` 3755
`\MT@checklist@` 864, 880, 1870
`\MT@checklist@family` 898
`\MT@checklist@font` 942
`\MT@checklist@size` 926
`\MT@checksetup` 3511, 3522, 3529, 3548, 3568
`\MT@clear@options` 190, 213, 298, 631, 3633
`\MT@clist@break` 488, 893, 921, 937, 951
`\MT@clist@function` 488
`\MT@cnt@encoding` 3123, 3131, 3132
`\MT@cnt@family` 3129, 3138, 3139
`\MT@cnt@series` 3136, 3145, 3146
`\MT@cnt@shape` 3143, 3151, 3152
`\MT@config@file` 3464, 3472, 3473, 3477, 3478, 3481
`\MT@context` 2008, 2020, 2039, 2066

- \MT@copy@font 787,
 815, 2932, 2938, 2950, 2957, 3388, 3390, 3753
 \MT@copy@font@ .. 815, 2932, 2938, 2950, 2957, 3388
 \MT@count 426, 427, 580, 1039, 1046, 1048,
 1049, 1054, 1059, 1060, 1064, 1072, 1103, 1165
 \MT@curr@file 1929, 1930,
 1939, 1940, 2722, 2723, 2905, 3068, 3477, 3955
 \MT@curr@list@name ... 1070, 1106, 1196, 1316,
 1881, 1910, 2077, 2217, 2225, 2230, 2238, 2244
 \MT@curr@ls 1566, 1649, 1652
 \MT@curr@set@name
 2454, 2456–2459, 2461, 2462, 2467, 2472,
 2476, 2477, 2502, 2506, 2535, 2565, 2571,
 2577, 2581, 2582, 2809, 2867, 2885, 2900,
 2912, 2976, 2980, 2985, 2988, 2990, 2994,
 2997, 3000, 3010, 3023, 3029, 3037, 3069, 3071
 \MT@declare@char@inh 3051, 3056, 3066
 \MT@declare@sets 2437, 2443, 2455, 2740
 \MT@DeclareSet 2432, 2434, 2451
 \MT@DeclareSetAndUseIt 2431, 2450
 \MT@DeclareVariants 2688, 2689, 2691
 \MT@def@bool@opt 3342,
 3355–3357, 3366, 3378, 3386, 3395, 3402
 \MT@def@n 305, 2425, 2426
 \MT@default@ex@set 2673
 \MT@default@kn@set 2673
 \MT@default@pr@set 2673
 \MT@default@sp@set 2673
 \MT@default@tr@set 2673
 \MT@define@code@key .. 2849, 2892–2895, 3091–3093
 \MT@define@code@key@font 2872, 2897, 3095
 \MT@define@code@key@size 2860, 2896, 3094
 \MT@define@opt@key .. 2898, 2914–2917, 3033–3035
 \MT@define@optionX 3501, 3557, 3558
 \MT@define@optionX@ 3559, 3587, 3589, 3590
 \MT@define@set@key@ 2465, 2621–2624
 \MT@define@set@key@font 2569, 2626
 \MT@define@set@key@size 2495, 2625
 \MT@detokenize@c 346, 2095, 2181
 \MT@detokenize@n 346, 2213
 \MT@dimen@six 974, 1040, 1064, 1103, 1721, 1724, 1850
 \MT@dinfo 86
 \MT@dinfo@list 877, 888, 891, 896, 905,
 908, 914, 916, 924, 932, 935, 940, 950, 953, 957
 \MT@dinfo@nl 86
 \MT@DisableLigatures 2731
 \MT@do@font 578, 990, 1282, 1413, 1483, 1560
 \MT@documentfalse 261
 \MT@documenttrue 261, 3899
 \MT@dofalse 856, 868, 892, 909, 936, 954
 \MT@dotrue 856, 859, 889, 906, 915, 933, 1865
 \MT@draftfalse 224, 3404
 \MT@drafttrue 224, 3406
 \MT@edef@n 307, 2044, 2070,
 2419, 2855, 2905, 2907, 3067, 3082, 3211, 3435
 \MT@encoding 846, 946,
 2003, 2015, 2095, 2181, 2212, 2229, 2236, 2244
 \MT@encoding@check 3080, 3084
 \MT@end@catcodes 1927, 1970, 2728,
 2742, 2769, 2791, 2810, 2824, 2838, 3061, 3476
 \MT@error 75, 288, 623, 699,
 1805, 1902, 1915, 2533, 2652, 2679, 2746,
 2924, 2941, 2966, 3380, 3397, 3534, 3552,
 3576, 3706, 3718, 3784, 3852, 3858, 3864, 3916
 \MT@ex@cc@name 1253,
 1255, 1322, 1323, 1333, 1339, 1340, 1352, 2775
 \MT@ex@context 818, 2386, 2424
 \MT@ex@doc@contexts 2424
 \MT@ex@factor 229, 1270, 1326
 \MT@ex@factor@ .. 1270, 1284, 1287, 1296, 1297, 1320
 \MT@ex@inh@name 1308–1310
 \MT@ex@level 229, 3748, 3750
 \MT@ex@max 241, 1299, 1300
 \MT@ex@min 241, 1302, 1303
 \MT@ex@setname 2644
 \MT@ex@split@val 1294
 \MT@exp@cs 302, 305, 308,
 311, 314, 829, 831, 840, 930, 1031, 1090,
 1093, 1123, 1127, 1137, 1140, 1143, 1310,
 1395, 1467, 2031, 2351, 2358, 2359, 2363,
 2405, 2416, 2501, 2866, 2883, 3113, 3186, 3229
 \MT@exp@gcs 302, 306, 310, 312, 315
 \MT@exp@one@n
 .. 317, 495, 619, 790, 792, 841, 884, 947,
 1506, 1934, 2332, 2339, 2369, 2402, 2413,
 2424, 2443, 2620, 2637, 2666, 2715, 3056, 3904
 \MT@exp@two@c
 .. 319, 337, 343, 353, 530, 789, 822, 824,
 826, 835, 1628, 1661, 1664, 2094, 2099, 2182
 \MT@exp@two@n 321, 902, 911, 2471, 2576
 \MT@expandfont 1257, 1273, 1277
 \MT@expansion 804, 1245, 3777
 \MT@expansionfalse 220, 3652, 3742
 \MT@expansiontrue 220, 3460
 \MT@extra@context 2424, 2761, 2775,
 2798, 2816, 2830, 2878, 2881, 2882, 2884,
 2918, 2933, 2939, 2951, 2956, 3042, 3183,
 3187, 3190, 3193, 3196, 3200, 3201, 3203, 3230
 \MT@extra@inputenc 3041, 3065, 3070, 3071
 \MT@factor@default 253, 3441, 3662
 \MT@family ... 790, 846, 903, 946, 2004, 2707, 2708
 \MT@familyalias
 .. 791, 792, 852, 910, 912, 2014, 2016, 2709
 \MT@feat 856, 979, 1012, 1085, 1087,
 1089, 1090, 1092, 1093, 1099, 1101, 1104,
 1112–1116, 1118, 1120–1124, 1127, 1128,
 1133, 1137, 1140, 1143, 1146–1148, 1157,
 1158, 1163, 1181, 1630, 1900, 1902, 1905,
 1906, 1910, 1912, 1915, 2040, 2042, 2044,
 2048, 2052, 2056, 2057, 2060, 2068, 2070,
 2071, 2073, 2080, 2084, 2218, 2219, 2248, 2250
 \MT@features 616,
 2424, 2437, 2620, 2631, 2660, 2902, 3051, 3485
 \MT@features@long 616, 619, 624, 2399, 3063
 \MT@file@list 1921,
 1923, 1931, 1934, 1936, 1941, 1944, 2715, 2719
 \MT@find@file 790, 792, 1921
 \MT@fix@catcode 5
 \MT@fix@font@set 2490, 3657
 \MT@font 316,
 789, 835, 842, 975, 982, 988, 989, 1020,
 1021, 1026, 1027, 1046, 1047, 1067, 1072,
 1077, 1171, 1174, 1207, 1208, 1210, 1211,
 1278, 1282, 1306, 1307, 1347, 1349, 1376,
 1377, 1383, 1384, 1390, 1391, 1400–1402,

- 1404, 1405, 1410–1412, 1455, 1456, 1462,
1463, 1472, 1473, 1475, 1476, 1481, 1482,
1506, 1510, 1565, 1874, 2252, 2303–2305,
2307, 2320, 2321, 2332, 2334, 2339, 2361, 2370
\MT@font@copy 818, 819, 826, 827, 835–837
\MT@font@list 2252, 2332–2334, 2393
\MT@font@orig 819
\MT@font@sets 2475, 2490, 2580, 3657
\MT@gdef@n 305, 2754, 2768,
2790, 2823, 2837, 2990, 3010, 3023, 3073, 3487
\MT@get@axis 2594–2597, 2607
\MT@get@basefamily 1933, 1971
\MT@get@basefamily@ 1977, 1980
\MT@get@char@unit 1011, 1135, 1161, 1249
\MT@get@charwd 1045, 1064, 1138, 1161
\MT@get@config 3464
\MT@get@ex@opt 1248, 1264, 1320
\MT@get@ex@opt@ 1328–1330, 1332, 1338
\MT@get@font 2575, 2585
\MT@get@font@ 2586, 2591, 2890
\MT@get@font@and@size 2877, 2889
\MT@get@font@dimen 1066, 1141
\MT@get@font@dimen@six . 962, 974, 1360, 1439, 1521
\MT@get@highlevel ... 2470, 2480, 2609, 2854, 3081
\MT@get@inh@list 965, 1251, 1363, 1442, 2066
\MT@get@listname 1986, 2042, 2068
\MT@get@listname@ 1986
\MT@get@ls@basefont 1534, 1651, 1658
\MT@get@opt 963, 1110, 1361, 1440
\MT@get@range 2499, 2509, 2864
\MT@get@size 2516, 2521, 2530, 2549, 2605
\MT@get@slot 1009, 1883, 2087, 3105, 3111
\MT@get@space@unit
.. 1135, 1235, 1374, 1381, 1388, 1453, 1460
\MT@get@tr@opt 1523, 1599
\MT@get@tr@opt@ 1617–1620, 1622
\MT@get@unit 1143, 1151, 1610
\MT@get@unit@ 1151
\MT@get@key 3620, 3635
\MT@glet 301, 312, 636,
822, 1661, 1681, 1807, 2301, 2394–2397,
2709, 2932, 2950, 3126, 3388, 3390, 3834, 3839
\MT@glet@nc 311, 1693, 1841, 2080,
2321, 2393, 2459, 2461, 2467, 2571, 2783,
2976, 2980, 2985, 2988, 2994, 2997, 3000, 3029
\MT@glet@nn 314, 2412, 3493
\MT@gobble@settings 973, 980
\MT@if@false 218,
683, 690, 707, 724, 2344, 3228, 3935, 3942
\MT@if@list@exists
... 961, 1247, 1263, 1359, 1438, 1522, 2039
\MT@if@true
218, 684–686, 691, 692, 708–710, 737, 741,
2337, 3237, 3241, 3249, 3254, 3936–3940, 3943
\MT@ifdefined@c@T
.. 325, 687, 713, 1029, 1053, 1308, 1393,
1465, 1553, 2169, 2707, 3070, 3173, 3497, 3500
\MT@ifdefined@c@TF
..... 325, 651, 1526, 1537, 1540, 1551,
1649, 1718, 1847, 1878, 1879, 2043, 2069, 3182
\MT@ifdefined@n@T
..... 325, 641, 881, 899, 927, 943, 1030,
1146, 1182, 1309, 1333, 1394, 1466, 1602,
1604, 1623, 2022, 2247, 2457, 2702, 2778,
2908, 3132, 3139, 3146, 3152, 3183, 3193, 3620
\MT@ifdefined@n@TF 325, 852,
862, 882, 900, 928, 944, 1112, 1120, 1322,
1339, 1868, 1905, 2010, 2095, 2646, 2649,
2675, 3159, 3210, 3219, 3283, 3486, 3490, 3902
\MT@ifdim .. 435, 550, 551, 555, 556, 2532, 2541,
3235–3237, 3239, 3240, 3247–3249, 3252, 3253
\MT@ifdimen 400, 2560, 2979, 2987, 2999, 3450
\MT@ifempty 355, 988, 989, 1018,
1024, 1224, 1225, 1373, 1380, 1387, 1410–
1412, 1424–1426, 1428–1430, 1452, 1459,
1481, 1482, 1498, 1499, 1699, 1700, 1710,
1728, 1817, 1852, 2404, 2436, 2441, 2482,
2488, 2510, 2511, 2524, 2525, 2599, 2610,
2630, 2635, 2659, 2664, 2737, 2805, 2899,
2904, 2918, 2931, 2939, 2949, 2956, 3008,
3050, 3055, 3085, 3296, 3327, 3466, 3508, 3565
\MT@ifint 368, 2148, 2806, 3009, 3434, 3443
\MT@ifstreq 442, 1188, 1901, 2041, 2056,
2198, 2201, 2368, 2574, 2876, 2978, 2996,
2998, 3174, 3299, 3301, 3304, 3307, 3329,
3331, 3345, 3346, 3412, 3413, 3418, 3422,
3449, 3510, 3517, 3521, 3528, 3567, 3573, 3876
\MT@in@clist 516, 619, 642, 884, 902, 911,
947, 1506, 1923, 1934, 2332, 2339, 2402, 2715
\MT@in@rlist 544, 930, 2031
\MT@in@rlist@ 544
\MT@in@rlist@@ 544
\MT@in@tlist 532, 1983, 2413
\MT@in@tlist@ 532
\MT@inannotfalse 98
\MT@inannottrue 98
\MT@increment 580, 3131, 3138, 3145, 3151
\MT@info 75, 90, 95, 3419,
3513, 3519, 3525, 3532, 3570, 3575, 3605, 3609
\MT@info@missing@char 1049, 1075
\MT@info@nl 75, 91, 92, 96, 652,
1076, 3410, 3420, 3473, 3491, 3494, 3655,
3661, 3668, 3749, 3778, 3795, 3801, 3807,
3812, 3819, 3824, 3855, 3861, 3867, 3912, 3923
\MT@inh@do 2079, 3096
\MT@inh@feat 3041, 3050, 3053, 3064
\MT@inh@split 3098, 3102
\MT@inlist@false 516, 520, 533, 545, 2347
\MT@inlist@true 516, 522, 540, 552, 557, 2347
\MT@is@active 2094, 2161
\MT@is@char 2099, 2182, 2188
\MT@is@composite 2097, 2209
\MT@is@feature 618, 2442, 2636, 2665
\MT@is@letter 2092, 2115, 2185, 2214
\MT@is@number 2131, 2136
\MT@is@symbol 2096, 2179
\MT@iterate 566
\MT@kerning 807, 1435, 3823
\MT@kerningfalse 226
\MT@kerningtrue 226
\MT@kn@c@name 1444, 1446, 1489, 2830
\MT@kn@context 2387, 2424
\MT@kn@doc@contexts 2424
\MT@kn@factor 229
\MT@kn@factor@ 1112
\MT@kn@inh@name 1465–1467

- `\MT@kn@max` 241
`\MT@kn@min` 241
`\MT@kn@setname` 2644
`\MT@kn@split@val` 1450
`\MT@kn@unit` 229
`\MT@kn@unit@` 1120, 1492
`\MT@led@kern` 648
`\MT@led@unhbox@line` 648
`\MT@led@mac@setup` 648, 705
`\MT@let@cn` 313, 314, 315, 853, 969, 1255,
1323, 1367, 1446, 1603, 1605, 1609, 1900,
1912, 2036, 2040, 2073, 2303, 2553, 2912, 3069
`\MT@let@enc` 311, 647, 1148, 2052, 2084,
2342, 3212, 3217, 3305, 3308, 3524, 3531, 3786
`\MT@let@nn`
314, 1113, 1118, 1121, 1133, 1340, 1343, 1624
`\MT@letterspace`
... 257, 1526, 1603, 1613, 3842, 3843, 3845
`\MT@letterspace@` 997, 1053, 1054,
1526–1528, 1531, 1535, 1539, 1547, 1559–
1561, 1566, 1576, 1724, 1728, 1816, 1853, 1856
`\MT@letterspace@default` 257, 3843
`\MT@listname`
1986, 2033, 2036, 2043, 2044, 2046, 2048,
2069, 2070, 2072, 2073, 2077, 2080, 3113, 3118
`\MT@load@inputenc` 1189, 1192, 1201
`\MT@load@list` 967, 1253, 1365, 1444, 1898
`\MT@loop` 566, 574, 3216
`\MT@lower` 2503, 2509, 2867
`\MT@ls@adjust` 1574, 1813, 1814
`\MT@ls@basefont` 1659–1661, 1664
`\MT@ls@outer@k`
... 1577, 1581, 1587, 1741, 1755, 1793, 1861
`\MT@ls@too@large` 1820, 1824, 3845
`\MT@lsfont` 1530, 1535,
1547, 1548, 1557, 1560, 1561, 1565, 1628,
1636, 1671, 1673, 1686, 1687, 1694, 1695,
1701, 1705, 1712, 1722, 1726, 1729, 1854, 1857
`\MT@lslig` 1648
`\MT@lua` 283, 372, 404, 446
`\MT@map@clist@` 488
`\MT@map@clist@c` ... 488, 828, 1882, 2338, 2350,
2357, 2392, 2399, 2437, 2439, 2631, 2633,
2660, 2662, 2902, 3051, 3053, 3063, 3485, 3618
`\MT@map@clist@n` 488, 861,
1867, 2424, 2468, 2497, 2572, 2620, 2692,
2752, 2852, 2862, 2874, 2992, 3006, 3108,
3291, 3294, 3323, 3325, 3354, 3429, 3506, 3563
`\MT@map@tlist@` 506
`\MT@map@tlist@c` 506, 535, 546, 831, 1031,
1310, 1395, 1467, 1990, 2363, 3184, 3229, 3657
`\MT@map@tlist@n` 506, 1962, 2190, 3125
`\MT@maybe@do` ... 856, 959, 1245, 1357, 1436, 1508
`\MT@maybe@etex` 262
`\MT@maybe@rem@from@list` 2363, 2367
`\MT@MT` 2, 75, 78, 81, 83, 85,
90, 91, 93, 94, 202, 209, 290, 630, 700, 702,
2941, 3467, 3605, 3609, 3656, 3880, 3916, 3917
`\MT@next@listname` 1993, 2002
`\MT@next@listname@` 2011, 2023, 2029
`\MT@n@ligatures` 1874, 2731
`\MT@n@setname` 1866, 2731
`\MT@n@ligatures` 809, 1863, 3891
`\MT@n@ligatures@` 1671, 1673, 1874, 1877
`\MT@n@ligatures@false` 223
`\MT@n@ligatures@true` 223, 2739
`\MT@nonselected@false` 1260, 1274
`\MT@nonselected@true` 1260, 1262
`\MT@noest@false` 2114, 2121, 2130, 2151, 2198, 2201
`\MT@noest@true` 2090, 2114, 2139
`\MT@old@cmd` 65, 70–73
`\MT@opt@autofalse` 3268
`\MT@opt@autottrue` 3268, 3356
`\MT@opt@def@set` 3282, 3310, 3335
`\MT@opt@DVIfalse` 3269
`\MT@opt@DVIttrue` 3269, 3359, 3362
`\MT@opt@expansion@false` 3267
`\MT@opt@expansion@true` 3267
`\MT@options` 2840
`\MT@optwarn@admissible` 3270, 3347, 3422
`\MT@optwarn@nan` 3274, 3436, 3445
`\MT@orig@add@accent` 2322
`\MT@orig@foreign@language` 3929, 3931
`\MT@orig@pickupfont` ... 730, 771, 776, 2257, 2324
`\MT@orig@py@macron` 768, 772, 777
`\MT@orig@select@language` 3924, 3926
`\MT@outer@kern` 1576, 1579, 1585, 1587,
1588, 1593, 1652, 1653, 1655, 1793, 1835, 1861
`\MT@outer@space`
... 1569, 1570, 1684, 1737, 1739, 1740,
1753, 1754, 1765, 1766, 1781, 1782, 1786, 1787
`\MT@pdf@annot` 98
`\MT@pdf@tex@no` 146, 196, 200, 275
`\MT@permute` 2767, 2789, 2803, 2822, 2836, 3076, 3122
`\MT@permute@` 3122
`\MT@permute@@` 3122
`\MT@permute@@@` 3122
`\MT@permute@@@@` 3150, 3155
`\MT@permute@@@@@` 3163, 3165
`\MT@permute@define` 3156, 3166–3168, 3208
`\MT@permute@reset` 3125, 3214
`\MT@permute@list`
... 2761, 2775, 2801, 2816, 2830, 2878,
2879, 2884, 3074, 3178, 3179, 3183, 3187,
3190, 3193–3196, 3203, 3204, 3230, 3261, 3262
`\MT@pickupfont` 773, 778, 2318, 2326
`\MT@plain` 123, 143
`\MT@pr@c@name` 967, 969, 1215, 2761
`\MT@pr@c@context` 818, 2386, 2424
`\MT@pr@doc@contexts` 2424
`\MT@pr@factor` 229, 3444, 3662, 3663
`\MT@pr@factor@` 1112
`\MT@pr@inh@name` 1029, 1030, 1032
`\MT@pr@level` 229, 3660, 3661
`\MT@pr@max` 241
`\MT@pr@min` 241
`\MT@pr@setname` 2644
`\MT@pr@split@val` 1016
`\MT@pr@unit` 229, 3451, 3664
`\MT@pr@unit@` 1120, 1218
`\MT@preset@aux` 1220,
1222, 1224, 1225, 1228, 1494, 1496, 1498, 1499
`\MT@preset@aux@factor` 1220, 1228, 1424–1426, 1494
`\MT@preset@aux@space` . 1222, 1228, 1428–1430, 1496
`\MT@preset@ex` 1334, 1351

- `\MT@preset@kn` 1487
`\MT@preset@kn@` 1487
`\MT@preset@pr` 1213
`\MT@preset@pr@` 1213
`\MT@preset@sp` 1417
`\MT@preset@sp@` 1417
`\MT@ProcessOptionsWithKV` 3613, 3636
`\MT@protrusion` 803, 959, 3667
`\MT@protrusionfalse` 219
`\MT@protrusiontrue` 219, 3457
`\MT@rbba@expansion` 611
`\MT@rbba@kerning` 611
`\MT@rbba@protrusion` 611
`\MT@rbba@spacing` 611
`\MT@rbba@tracking` 611
`\MT@register@font` 812, 2334, 2395
`\MT@register@font@cx` 2356, 2395
`\MT@register@subst@font` 2309, 2333, 2396
`\MT@register@subst@font@cx` 2349, 2396
`\MT@rem@from@clist` 527, 841, 2369, 2844
`\MT@rem@from@list` 831, 839
`\MT@repeat` 566, 576, 3222
`\MT@requires@latex` 142, 191, 633, 726, 755, 1627, 2255, 3275, 3428, 3613, 3617
`\MT@requires@luatex` 278, 370, 402, 444, 2922, 3377, 3560, 3793, 3804, 3815
`\MT@requires@pdftex` 274, 368, 400, 442, 648, 786, 795, 806, 808, 816, 995, 1037, 1051, 1285, 1356, 1435, 1503, 1668, 1863, 2731, 2920, 3022, 3028, 3375, 3459, 3559, 3703, 3728, 3781, 3871, 3888
`\MT@res@a` 465, 467, 518, 525, 528, 530, 534, 539
`\MT@res@b` 466, 467, 528–530, 538, 539
`\MT@reset@context` 2377, 2380, 2384, 2411
`\MT@reset@context@` 2384, 2411
`\MT@reset@ef@codes` 1250, 1272, 1285, 1335
`\MT@reset@ef@codes@` 1280, 1288, 1292
`\MT@reset@kn@codes` 1441, 1448, 1478
`\MT@reset@kn@codes@` 1478
`\MT@reset@pr@codes` 964, 971, 992
`\MT@reset@pr@codes@` 992
`\MT@reset@sp@codes` 1362, 1369, 1407
`\MT@reset@sp@codes@` 1407
`\MT@restore@catcodes` ... 5, 7, 8, 213, 298, 631, 3957
`\MT@restore@p@h` 681, 693, 742
`\MT@saved@setupfont` 3603, 3610
`\MT@scale` 587, 1040, 1086, 1100, 1103, 1297
`\MT@scale@factor` 1042, 1061, 1084, 1230, 1353
`\MT@scale@to@em` .. 1019, 1025, 1037, 1058, 1236, 1375, 1382, 1389, 1454, 1461, 1612, 1719, 1848
`\MT@selectedfalse` 222
`\MT@selectedtrue` 222
`\MT@series` 846, 946, 2005, 2017
`\MT@set@all@ex` 1280, 1354
`\MT@set@all@kn` 1478, 1500
`\MT@set@all@pr` 985, 992, 1226
`\MT@set@all@sp` 1407, 1432
`\MT@set@babel@context` 3901, 3927, 3932, 3946
`\MT@set@codes` .. 970, 1000, 1256, 1368, 1447, 1913
`\MT@set@curr@ls` 1566
`\MT@set@curr@ok` 1579, 1580, 1593, 1770
`\MT@set@curr@os` 1568, 1771
`\MT@set@default@set` 2660, 2666, 2673
`\MT@set@ex@codes` 1276, 3733, 3735
`\MT@set@ex@codes@n` 1260, 1276, 3735
`\MT@set@ex@codes@s` 1246, 3733
`\MT@set@ex@heirs` 1310, 1346
`\MT@set@inh@list` 3047, 3049
`\MT@set@inputenc` 966, 1179, 1252, 1364, 1443, 2078
`\MT@set@inputenc@` 1182, 1184
`\MT@set@kn@codes` 1437
`\MT@set@kn@heirs` 1467, 1471
`\MT@set@listname` 968, 1111, 1254, 1321, 1366, 1445, 1601, 2217
`\MT@set@lsfont` 1564, 1627, 1636
`\MT@set@named@keys` 2763, 2777, 2799, 2818, 2832, 2840
`\MT@set@pr@codes` 960
`\MT@set@pr@heirs` 1033, 1206
`\MT@set@sp@codes` 1358
`\MT@set@sp@heirs` 1395, 1399
`\MT@set@tr@codes` 1518, 1631, 1641
`\MT@SetExpansion` 2773, 2775
`\MT@SetExtraKerning` 2828, 2830
`\MT@SetExtraSpacing` 2814, 2816
`\MT@SetProtrusion` 2759, 2761
`\MT@SetTracking` 2795, 2797
`\MT@setup@` 632, 635, 636, 3369, 3370
`\MT@setup@contexts` 2376, 2391
`\MT@setupfont` .. 785, 2312, 3603, 3606, 3610, 3644
`\MT@setupfont@hook` 682, 706, 712, 717, 742, 749, 785
`\MT@shape` 846, 946, 2006, 2018
`\MT@shorthandoff` 3910, 3941, 3944
`\MT@show@pdfannot` 104
`\MT@shrink` 229, 1267, 3674, 3675, 3678, 3679, 3682, 3686, 3738, 3751
`\MT@shrink@` 1267, 1278, 1320
`\MT@shrink@default` 254
`\MT@size` 551, 555, 556, 846, 946
`\MT@size@name` 544, 2033
`\MT@sp@c@name` 1365, 1367, 1419, 2816
`\MT@sp@context` 2387, 2424, 3876
`\MT@sp@doc@contexts` 2424
`\MT@sp@factor` 229
`\MT@sp@factor@` 1112
`\MT@sp@inh@name` 1393–1395
`\MT@sp@max` 241
`\MT@sp@min` 241
`\MT@sp@setname` 2644
`\MT@sp@split@val` 1371
`\MT@sp@unit` 229
`\MT@sp@unit@` 1120, 1422
`\MT@spacing` 807, 1356, 3811
`\MT@spacingfalse` 225
`\MT@spacingtrue` 225
`\MT@split@codes` 1002, 1006
`\MT@split@name` 789, 846
`\MT@step` 229, 1268, 3677, 3693, 3700, 3752, 3757, 3758, 3762
`\MT@step@` 1268, 1278, 1320
`\MT@step@default` 254
`\MT@stretch` 229, 1266, 3671, 3672, 3675, 3678, 3680, 3685, 3688, 3737, 3751
`\MT@stretch@` 1266, 1278, 1320
`\MT@stretch@default` 254, 3672
`\MT@temp` 987–990, 1409–1413, 1480–1483, 1689, 1693, 1696,

- 1701, 1704, 1709, 1765, 1772, 1775, 1778,
1781, 1786, 1793, 1796, 1838, 1841, 1858,
2593, 2618, 3616, 3621, 3626, 3627, 3629, 3632
- \MT@tempencoding 3158, 3159, 3169, 3175
- \MT@tempfamily 3170
- \MT@tempseries 3171
- \MT@tempshape 3172
- \MT@tempsize 3126, 3173, 3182, 3184, 3188
- \MT@test@ast 1717, 1846, 2481, 2486
- \MT@textls 49, 1813, 1814, 1816
- \MT@the@pr@code 994, 1020, 1026, 1562
- \MT@the@pr@code@tr 994, 1562
- \MT@tlist@break 506, 541, 563, 1995, 3264
- \MT@toks 217, 1076, 1106, 1316,
2091, 2119, 2126, 2173, 2203, 2222, 2234, 2242
- \MT@tr@cc@name
... 1602–1605, 1609, 1623, 1624, 2800, 2807
- \MT@tr@context 2387, 2424
- \MT@tr@doc@contexts 2424
- \MT@tr@factor@ 1611
- \MT@tr@font@list 1504
- \MT@tr@ispace 1537, 1538, 1617
- \MT@tr@ligatures 1553, 1620, 1670, 1673
- \MT@tr@max 241, 1825–1827
- \MT@tr@min 241, 1829–1831
- \MT@tr@noligatures 1553, 1667
- \MT@tr@okern 1551, 1619
- \MT@tr@ospace 1540, 1541, 1617
- \MT@tr@outer@icr 1775, 1798
- \MT@tr@outer@icr@ 1798
- \MT@tr@outer@l 1572, 1733
- \MT@tr@outer@next 1761, 1800
- \MT@tr@outer@r 1595, 1761, 1798
- \MT@tr@outer@r@ 1761
- \MT@tr@set@okern 1552, 1835
- \MT@tr@set@okern@ 1839, 1840, 1845
- \MT@tr@set@space 1543, 1685
- \MT@tr@set@space@ 1690–1692, 1698
- \MT@tr@set@space@@ 1703, 1708, 1711, 1716
- \MT@tr@set@name 2644
- \MT@tr@unit@ 1604, 1718, 1847
- \MT@tracking 796, 1504, 1631, 2314, 3587, 3588, 3800
- \MT@tracking@ 1504, 3588
- \MT@trackingfalse 227
- \MT@trackingtrue 227
- \MT@try@order 1990, 1998
- \MT@undefined@char 2166, 2168, 2178
- \MT@upper 2503, 2509, 2867
- \MT@use@set 2631, 2637, 2644
- \MT@val 2404–2406, 2408, 2414, 2416,
2419, 2469, 2473, 2481, 2483, 2498–2500,
2512, 2515, 2517, 2520, 2522, 2523, 2529,
2531, 2532, 2534, 2537, 2539, 2550, 2553,
2557, 2560–2562, 2564, 2566, 2573–2575,
2578, 2587–2589, 2601, 2603, 2608, 2610,
2612, 2614, 2853, 2855, 2863–2865, 2875–
2877, 2880, 2885, 3079, 3080, 3082, 3086,
3107, 3113, 3117, 3118, 3283, 3284, 3287,
3295, 3296, 3299, 3301, 3304, 3307, 3326,
3327, 3329, 3331, 3507, 3508, 3510, 3517,
3521, 3528, 3534, 3564, 3565, 3567, 3573, 3576
- \MT@variants 1983, 2685, 2695
- \MT@vinfo 75, 92, 802, 874, 1115, 1124,
1128, 1163, 1324, 1341, 1520, 1528, 1885,
1894, 1906, 1930, 1940, 1943, 2046, 2048,
2054, 2385, 2408, 2717, 2723, 3410, 3417, 3903
- \MT@warn@ascii 2128, 2221
- \MT@warn@axis@empty 2600, 2611, 2616
- \MT@warn@code@too@large 1090, 1093, 1097
- \MT@warn@err 75, 3414, 3415
- \MT@warn@ex@too@large 1300, 1303, 1314
- \MT@warn@lua 3783, 3793, 3804, 3815
- \MT@warn@maybe@inputenc ... 1880, 2235, 2243, 2246
- \MT@warn@nodim 2971, 2981, 2989, 3001
- \MT@warn@number@too@large 2155, 2227
- \MT@warn@preset@t@width ... 1219, 1239, 1423, 1493
- \MT@warn@rest 2108, 2232
- \MT@warn@tracking@DVI 1529, 3827
- \MT@warn@unknown 2105, 2240
- \MT@warning 66, 75, 93, 1826,
1830, 1888, 2057, 2458, 2564, 2617, 2703,
2725, 2808, 2909, 2972, 3012, 3025, 3087,
3176, 3194, 3261, 3414, 3419, 3497, 3595–3597
- \MT@warning@nl
... 75, 87, 94, 199, 289, 629, 664, 673, 719,
976, 1068, 1104, 1156, 1196, 1240, 1315,
1678, 2222, 2228, 2233, 2241, 2285, 2780,
2958, 3271, 3277, 3287, 3415, 3420, 3452,
3480, 3641, 3695, 3739, 3760, 3829, 3877, 3948
- \MT@while@enum 572, 578, 745–747
- \MT@with@babel@and@T
... 640, 684–686, 708–710, 3936–3940, 3943
- \MT@with@package@T 638, 688, 691, 692,
698, 705, 715, 727, 741, 743, 756, 767, 2296
- \MT@xadd 102, 474, 1931, 1936, 1941,
1944, 2351, 2359, 2416, 2501, 2719, 2866, 3113
- \MT@xaddb 481, 2883, 3186
- \MT@xdef@n .. 307, 2647, 2650, 2677, 2682, 2706,
2807, 2878, 2900, 3037, 3071, 3203, 3284, 3286
- ## N
- \normalfont 784, 3770, 3774
- \nullfont 784
- ## P
- \pdfadjustinterwordglue 3589, 3806
- \pdfadjustspacing 3558, 3748, 3773
- \pdfappendkern 3591, 3818
- \pdfcopyfont 826
- pdfcpot (package) 4, 27, 29, 50, 59, 87, 179
- \pdffontexpand 1278
- \pdfnoligatures 1887, 1893
- \pdfprependkern 3590, 3817
- \pdfprotrudechars 3557, 3660, 3797, 3838
- \pdftracingfonts 277
- \pickup@font 730, 771, 773,
776, 778, 2284, 2286, 2295, 2318, 2324, 2326
- pifont (package) 87
- pinyin (package) 52, 184
- protrusion (option) 6, 7, 11, 111, 112
- PSNFSS (package) 175
- pstricks (package) 9, 119
- pxfonts (package) 21, 127

Q	
qfonts (package)	21, 127, 182
R	
ragged2e (package)	74, 185
reysize (package)	95
\rightmarginkern	657
\rpxcode	989, 1026, 1027, 1208, 1211, 1561
S	
selected (option)	8, 14, 32, 64, 81, 112, 180
\selectfont	1632, 2379, 2389, 3954
\set@fontsize	2557
\SetExpansion	14, 36, 2771
\SetExtraKerning	19, 38, 2826
\SetExtraSpacing	18, 39, 2812
\SetProtrusion	13, 35, 2757
\SetTracking	16, 37, 2793
\sfcode	3875
\shbscode	1390, 1391, 1402, 1405, 1412
\showhyphens	3768, 3772
shrink (option)	8, 16, 32, 114, 120, 122, 182, 184
slantsc (package)	154
soul (package)	4, 29, 51, 52, 75, 183, 185
soulutf8 (package)	29, 185
\spacefactor	1744
\spaceskip	1745, 1748
spacing (option)	7, 11, 30, 112
\stbscode	1383, 1384, 1401, 1405, 1411
step (option)	8, 32, 114, 120, 122, 180, 182, 184
stretch (option)	8, 16, 25, 32, 114, 120, 122, 184
T	
\tagcode	1557, 1879, 1884
U	
tex4ht (package)	50, 183
\textls	24, 52, 53, 735, 758, 763, 1575, 1644, 1811
\textls*	24
\textmicrotypecontext	22, 48, 731, 2383
tipa (package)	87
trace (package)	88, 183
\tracingmicrotype	86
\tracingmicrotypeinpdf	97
\tracingmicrotypeinpdfallfalse	103
\tracingmicrotypeinpdfalltrue	103
tracking (option)	7, 11, 24, 30, 112
txfonts (package)	21, 127
V	
ucs (package)	84, 185
ulgothic (package)	21
unit (option)	8, 31, 181–183
\UseMicrotypeSet	11, 33, 60, 72, 2452, 2628
V	
verbose (option)	9, 29, 31, 114, 180, 181, 185
W	
wordcount (package)	48, 184
X	
\xspaceskip	1747, 1750
Y	
yfonts (package)	24
Z	
zefonts (package)	21, 127

C The L^AT_EX Project Public License

LPPL Version 1.3c 2006-05-20

Copyright 1999, 2002–2006 L^AT_EX3 Project

Preamble

The L^AT_EX Project Public License (LPPL) is the primary license under which the the L^AT_EX kernel and the base L^AT_EX packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is T_EX-related (such as a L^AT_EX package), but it is written in such a way that you can use it even if your work is unrelated to T_EX.

The section ‘WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE’, below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the L^AT_EX3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document ‘cfigguide.tex’ and ‘modguide.tex’ in the base L^AT_EX distribution for suggestions.

Definitions

In this license document the following terms are used:

Work: Any work being distributed under this License.

Derived Work: Any work that under any applicable law is derived from the Work.

Modification: Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

Modify: To apply any procedure that produces a Derived Work under any applicable law.

Distribution: Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun's Network File System (NFS).

Compiled Work: A version of the Work that has been processed into a form where it is directly usable on a

computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

Current Maintainer: A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the 'Copyright Holder' under any applicable law.

Base Interpreter: A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a 'L^AT_EX-Format' or in the case of files belonging to the 'L^AT_EX-format' a program implementing the 'T_EX language'.

Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.
2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.
5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
 - (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
 - (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.
 - (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
 - (d) You distribute at least one of the following with the Derived Work:

- i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
 - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
 8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
 9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.
 10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
(b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.
 11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
 12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any au-

thor named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work

by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
 - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
 - (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent

- community your intention to take over maintenance. (If the Work is a L^AT_EX work, this could be done, for example, by posting to `comp.text.tex`.)
3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.
 - (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.
 4. If you make an ‘intention announcement’ as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.
 5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document ‘`modguide.tex`’ in the base L^AT_EX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing L^AT_EX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L^AT_EX, the discussion in ‘`modguide.tex`’ may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
% http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status ‘maintained’.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the ‘Work’ referring to the three files ‘`pig.dtx`’, ‘`pig.ins`’, and ‘`pig.sty`’ (the last being generated from ‘`pig.dtx`’ using ‘`pig.ins`’), the ‘Base Interpreter’ referring to any ‘L^AT_EX-Format’, and both ‘Copyright Holder’ and ‘Current Maintainer’ referring to the person ‘M. Y. Name’.

If you do not want the Maintenance section of LPPL to apply to your Work, change ‘maintained’ above into ‘author-maintained’. However, we recommend that you use ‘maintained’ as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case

(e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

Important Recommendations

Defining What Constitutes the Work

The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which

files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.