

The microtype package

Subliminal refinements towards typographical perfection

R Schlicht
w.m.1@gmx.net

v2.8c
2021/03/14

The microtype package provides a \LaTeX interface to the micro-typographic extensions that were introduced by pdf\TeX and have since also propagated to Lua\TeX and Xe\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. These features may be applied to customisable sets of fonts, and all micro-typographic aspects of the fonts can be configured in a straight-forward and flexible way. Settings for various fonts are provided.

Note that character protrusion requires pdf\TeX (version 0.14f or later), Lua\TeX , or Xe\TeX (at least version 0.9997). Font expansion works with pdf\TeX (version 1.20 for automatic expansion) or Lua\TeX . The package will by default enable protrusion and expansion if they can safely be assumed to work. Disabling ligatures requires pdf\TeX (≥ 1.30) or Lua\TeX , while the adjustment of interword spacing and of kerning only works with pdf\TeX (≥ 1.40). Letterspacing is available with pdf\TeX (≥ 1.40) or Lua\TeX (≥ 0.62).

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 © 2004–2021 R Schlicht. It may be distributed and/or modified under the conditions of the [\$\text{\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’.

Contents

1	Micro-typography with T_EX	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	7
3.4	Tracking	8
3.5	Miscellaneous options	9
3.6	Changing options later	9
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	15
5.4	Additional kerning	18
5.5	Interword spacing	19
5.6	Character inheritance	20
5.7	Configuration files	20
6	Context-sensitive setup	22
7	Letterspacing revisited	23
8	Disabling ligatures	24
9	Hints and caveats	25
10	Contributions	28
11	Acknowledgments	28
12	References	30
13	Short history	30
14	Implementation	34
14.1	Preliminaries	35
	Debugging [36] Requirements [38] Declarations [42] Auxiliary macros [43] Compatibility [52]	
14.2	Font setup	56
	Protrusion [62] Expansion [69] Interword spacing (glue) [72] Additional kern- ing [73] Tracking [75] Disabling ligatures [85] Loading the configuration [87] Translating characters into slots [91] Hook into L ^A T _E X's font selection [99] Context-sensitive setup [102]	
14.3	Configuration	105
	Font sets [105] Variants and aliases [111] Disabling ligatures [112] Interaction with babel [113] Fine tuning [113] Character inheritance [120] Permuta- tion [121]	

14.4	Package options	125
	Declaring the options [125] Loading the definition file [129] Reading the configuration file [129] Hook for other packages [130] Changing options later [131] Processing the options [133]	
15	Configuration files	143
15.1	Font sets	143
15.2	Font variants and aliases	144
15.3	Interaction with babel	146
15.4	Note on admissible characters	147
15.5	Character inheritance	147
	OT1 [147] T1 [148] LY1 [148] OT4 [149] QX [149] T5 [150] EU1, EU2, TU [151] Euro symbols [151]	
15.6	Tracking	152
15.7	Font expansion	152
15.8	Character protrusion	154
	Normal [155] Italics [163] Small caps [174] Italic small caps [177] Text companion [178] Computer Modern math [183] AMS symbols [187] Euler [190] Euro symbols [194]	
15.9	Interword spacing	195
	Nonfrenchspacing [197]	
15.10	Additional kerning	198
	French [199] Turkish [199]	
16	OpenType configuration files	201
16.1	Character inheritance	201
16.2	Character protrusion	205
17	Auxiliary file for micro fine tuning	219
A	The title logo	221
A.1	Macros	221
A.2	Document	226
B	The letterspacing illustration	226
B.1	Macros	226
B.2	Document	229
C	Change history	231
D	Index	241
E	The L^AT_EX Project Public License	250

List of Tables

1	Availability of micro-typographic features	7
2	Predefined font sets	11
3	Fonts with tailored protrusion settings	21
4	Order for matching font attributes	89

1 Micro-typography with T_EX

Micro-typography is the art of enhancing the appearance and readability of a document while exhibiting a minimum degree of visual obtrusion. It is concerned with what happens between or at the margins of characters, words or lines. Whereas the macro-typographical aspects of a document (i.e., its layout) are clearly visible even to the untrained eye, micro-typographical refinements should ideally not even be recognisable. That is, you may think that a document looks beautiful, but you might not be able to tell exactly why: good micro-typographic practice tries to reduce all potential irritations that might disturb a reader.

Some essential micro-typographical aspects are already taken care of by T_EX out of the box – and in an outstanding manner – namely, hyphenation and justification, as well as kerning and ligatures. Other aspects are in the user’s scope of responsibilities, e.g., to specify the right amounts of spacing around punctuation characters, numbers, or quotation marks. On top of this, a number of long-standing micro-typographic techniques have been introduced to the T_EX world relatively recently with pdfT_EX, and have since also propagated to LuaT_EX and X_YT_EX. These features make them 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, the author of pdfT_EX, who writes in his 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]

Another micro-typographic technique, which has always been extremely difficult to achieve in T_EX, is robust and hyphenatable *letterspacing* (*tracking*).¹ Whereas letterspacing can easily be, and often is, abused when applying it to lowercase letters, readability may be increased by slightly letterspacing (small) capitals or by decreasing the tracking of very large uppercase type.

Setting *additional kerning* for individual characters is especially (but not only) 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

¹ The `soul` package undertakes great efforts, but may still fail in certain circumstances; even to systematically adjust the tracking of a font throughout the document remains impossible.

by making these characters active (as is done, for example, by the `babel` package), which may not always be a robust solution. In contrast to the standard kerning built into the fonts (which will of course apply as usual), this additional kerning relates to single characters, not to character pairs.

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 TeX’s ‘space factors’. 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 influence the interword space. Also, the settings shipped with `microtype` are but a first approximation, and I would highly welcome corrections and improvements. I suggest reading the reasoning behind the settings in section 15.9.

The possibility, finally, to *disable all or selected ligatures* is particularly useful for typewriter 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 and systematic manner. The next chapters present a survey of all options and customisation possibilities. Should the micro-typographic extension discussed in a section work only with certain TeX engines, this requirement is marked inside a grey text box on the right.

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 (however unlikely this would seem, 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 desired micro-typographic features, 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’. A number of 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 will 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 \LaTeX packages, the `microtype` package accepts options in the well-known key=value syntax. In the following, you will 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 – if preceded by an asterisk, this default only applies when running an up-to-date \pdfTeX in PDF 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 always be enabled, font expansion will only be disabled when the fonts cannot be expanded automatically, that is, with \pdfTeX versions older than 1.20 or in DVI output mode (see section 3.5), or with \XeTeX . In other words, `microtype` will try to apply as much micro-typography as can safely be expected to work under the respective conditions (hence, it is usually not necessary to load the package with different options, e.g., for PDF resp. DVI mode).

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

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

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

```
\usepackage{microtype}
```

With activated font expansion and/or 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 true, false, ** false

This option will systematically change the tracking of the fonts specified in the active font set (by default, all small capitals). It is not available with \XeTeX (you may use the ‘LetterSpace’ option of the `fontspec` package instead). With \pdfTeX , it is only available in PDF mode.

kerning true, false, ** false

spacing These features do not unconditionally improve the quality of the typeset text: the spacing feature is still considered experimental, while the kerning feature only makes sense in special cases. Therefore, neither feature is enabled by default. They are not available with \XeTeX or \LuaTeX .

Table 1:

Availability of micro-typographic features

T _E X engine			Micro-typographic features					
Engine	Version	Output	Protrusion	Expansion		Tracking	Kerning	Spacing
				manual	automatic			
pdfT _E X	< 0.14f	DVI/PDF	∅	∅	∅	∅	∅	∅
	≥ 0.14f	DVI/PDF	★	☒	∅	∅	∅	∅
	≥ 1.20	DVI	★	☒	∅	∅	∅	∅
		PDF	★	☒	★	∅	∅	∅
	≥ 1.40	DVI	★	☒	∅	∅	☒	☒
		PDF	★	☒	★	☒	☒	☒
LuaT _E X	≥ 0.30	DVI	★	☒	∅	∅	∅	∅
		PDF	★	☒	★	∅	∅	∅
	≥ 0.62	DVI	★	∅	(☒) ^a	☒	∅	∅
		PDF	★	∅	★	☒	∅	∅
X _Y T _E X	≥ 0.9997	PDF	★	∅	∅	∅	∅	∅
★ = enabled ☒ = not enabled ∅ = not available ^a by means of variable tracking								

Table 1 presents an overview of which micro-typographic features are available and enabled by default for the relevant T_EX 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

pdfT_EX 0.14f | LuaT_EX 0.30 | X_YT_EX 0.9997

factor

`<integer>`

1000

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

unit

character, `<dimension>`

character

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

3.3 Font expansion

pdfT_EX 0.14f | LuaT_EX 0.30

auto

`true`, `false`

* `true`

Beginning with pdfT_EX version 1.20 (inherited by LuaT_EX), the expanded instances of the fonts may be calculated automatically and at run-time instead of the user

having to prepare them in advance. This option is true by default provided that you are using a TeX engine with this capability and the output mode is PDF.² If auto is set to false, the font instances for all expansion steps must exist (with files called *(font name)±(expansion value)*, e.g., cmr12+10, as described in the [pdfTeX manual](#)). With recent versions of LuaTeX, expansion is always automatic.

When generating DVI files, font expansion has to be enabled explicitly. With pdfTeX, *automatic* font expansion will not work because the postprocessing drivers (dvips, dvipdfm, etc.) resp. the DVI viewer are not able to generate the fonts on the fly. With LuaTeX, on the other hand, expansion in DVI mode is realised by modifying the inter-letter spacing (tracking) instead of the glyphs themselves, which may or may not be desired.

stretch *(integer)* 20

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

step *(integer)* * 1

Fonts are not expanded by arbitrary amounts but only by certain discrete steps within the expansion limits. With recent versions of pdfTeX (1.40 or newer) or LuaTeX, this option is by default set to 1, in order to allow trying the maximum number of font instances, and hence to guarantee the best possible output.³ Older pdfTeX versions, however, had to include every font instance in the PDF file, which may increase the file size quite dramatically. Therefore, in case you are using a pre-1.40 pdfTeX version, step is by default set to one fifth of the smaller value of stretch and shrink.

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 increasing 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.

3.4 Tracking

pdfTeX 1.40 | LuaTeX 0.62

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.

- 2 With pdfTeX, automatic font expansion does not work with bitmap fonts. Therefore, if you are using the Computer Modern Roman fonts in T1 encoding, you should either install the cm-super package or use the Latin Modern fonts (package lmodern).
- 3 The downside with this default is that pdfTeX may run out of memory with huge documents; in this case, read about the error messages in the ‘Hints and caveats’ section (9), or try with a larger step.

3.5 Miscellaneous options

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=microtype`.

DVIoutput true, false * false

`pdfTeX` and `LuaTeX` are not only able to generate PDF output but can also spit out DVI files. In fact, all recent `TeX` systems are using `pdfTeX` as the default engine also for DVI output, and `LuaTeX` too can be called in DVI mode. However, since changing the output mode inside the document may have undesired effects, this option should be considered deprecated; instead, it is recommended to just call the respective program (`latex` resp. `dvilualatex`). For `XYTeX`, this option is not applicable.

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**, **Kerning** and **spacing** 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 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. In the main configuration file `microtype.cfg`, a font set called ‘`basictext`’ is defined as follows:

```
\DeclareMicrotypeSet{basictext}
{ encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU},
  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, 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,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU} }
```

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 be translated into the document’s `\<value>default`, e.g., `\rmdefault`.⁴ A single asterisk means `\<attribute>default`, e.g., `\encodingdefault`, respectively `\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

4 These translations will take place `\AtBeginDocument`, which means that changes to the defaults inside the preamble will also be taken into account. Only in cases where you change font defaults `\AtBeginDocument` yourself, you need to load `microtype` after these changes.

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
alltext-nott (allmath-nott)	Text encodings, TS1 (OML , OMS , U)	\rm* , \sf*	Ø	Ø	Ø
basictext (basicmath)	Text encodings (OML , OMS)	\rm* , \sf*	\md*	Ø	\normalsize , \footnotesize , \small , \large
smallcaps	Text encodings	Ø	Ø	\sc* , si , scit	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 , T2A , LY1 , OT4 , QX , T5 , EU1 , EU2 , TU ‘\...*’ = ‘\...default’					

boundary is included in the range, the upper boundary is not. Thus, ‘12-16’ would match 12 pt, 13.5 pt and 15.999 pt, for example, but not 16 pt. You are allowed to omit the lower or upper bound (‘-10’, ‘large-’).

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

```
\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 permitted 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 eleven predefined font sets. They may also be activated by passing their name to the feature options protrusion, expansion, tracking, kerning and spacing when loading the package, for example:

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

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

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

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

If a feature is enabled but no font set has been chosen explicitly, the sets declared by this command will be activated. By default, the ‘alltext’ font set will be activated for character protrusion and additional kerning, the ‘alltext-nott’ 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, kerning or spacing 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. Here, as in all configuration commands, all spaces are ignored.

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), with the only difference that values including asterisks (which, as you may recall, stand for the respective default) will be translated 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 settings exist for both the current family (say, T1/cmr//) and for italic fonts in the normal weight (T1/m/it/), the settings for the cmr family would apply. The encoding must always match.

The characters may be specified either as a single letter (A), as a text symbol command (`\textquoteleft`), or as a slot number (resp. Unicode number for LuaTeX or XeTeX): three or more digits for decimal notation, prefixed with “ for hexadecimal, with ‘ for octal numerals (e.g., the ‘fl’ ligature in T1 encoding: 029, “1D, ‘35). 8-bit (and even UTF-8) characters may be entered directly or in L^AT_EX’s traditional 7-bit notation: both “A and Ä are valid, provided the character is actually declared in both the input and the font encoding. With LuaTeX or XeTeX, you may additionally specify a (font-specific) glyph name, prefixed with ‘/’ (e.g., the ‘fl’ ligature as /f_l). Note that you also have the possibility to declare lists of characters that should inherit settings (see section 5.6).

5.1 Character protrusion

pdfTeX 0.14f | LuaTeX 0.30 | XeTeX 0.9997

\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 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 may 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 assigned a name to it) before the current list will be loaded, so that the fonts will inherit the values from the loaded list.

In this way, 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, in other words, any options from the loaded lists will be ignored:

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

inputenc 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 further details, see section 6.

5.2 Font expansion

pdfTeX 0.14f | LuaTeX 0.30

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

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

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.

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 be avoided by shrinking the font a 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 a `fussy' widow.}
```

This method of employing contexts to temporarily apply different expansion parameters only works with pdfTeX version 1.40.4 or later,⁶ or with LuaTeX. Also note that both pdfTeX and LuaTeX prohibit the use of fonts with different expansion limits or steps (even of different fonts) within one paragraph, hence the `sloppy` context would have 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

pdfTeX 1.40 | LuaTeX 0.62

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

An important typographic technique – which was missing in TeX for a long time – is the adjustment of tracking, i.e., the uniform addition or subtraction of letter space

⁶ For older versions, a dirty trick is laid out in section 14.2 on page 59.

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.⁷ The `\SetTracking` command allows specifying 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 default equal to the current letterspace amount. To remove kerning on both sides, you would write 'outer kerning={0,0}'.

no ligatures By default, ligatures in letterspaced fonts will 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. With pdfTeX, this is not recommended, however, since it entails that kerning will be switched off, too. With LuaTeX, there is no such limitation. The default settings disable ligatures for the character 'f' only, i.e., 'ff',

⁷ 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.

‘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 \slig (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 are in no way recommended; they only serve 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 would be the (typographically dubious) outcome:

Stop stealing sheep!

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.

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.

As another, more realistic example, suppose you want to space out all small capitals by $50/1000\text{em}$, fonts smaller than \small by 0.02em, and to decrease the tracking of large type by 0.02em. This could be achieved 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

⁸ With pdfTeX 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.

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 Additional kerning

pdfTeX 1.40

`\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. (Put differently, this feature allows modifying the left or right *sidebearings* of specific glyphs.)

It should not be neglected to mention a limitation of this feature: 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'. Furthermore, additional kerning will not be applied in math mode. These restrictions of pdfTeX will hopefully be lifted some time.

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 1 em.

context When it comes to kerning settings, this option is especially useful, since it allows applying 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}
```

5.5 Interword spacing

pdfTeX 1.40

`\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, T_EX 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 T_EX 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, T_EX has the concept of ‘space factors’. They may be used to increase the space after certain characters, most prominently the punctuation characters. pdfT_EX’s additional spacing adjustment may be considered as an extension to space factors with much finer control: 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 – you may 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. Note that when interword spacing adjustment is in effect, space factors are ignored.

The *spacing settings* are declared as pairs of *character* = *spacing factors*, 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, but the settings must always 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 the following (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 \fontdimen 4). Conversely, setting all three values to -1000 would completely cancel a space after the respective character.

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. 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.

The situation is different with LuaTeX and XeTeX, however: the default inheritance settings only contain those glyphs that can safely be assumed to exist in any font; but since OpenType fonts may contain many more glyphs for different scripts (languages), it is quite probable that font-specific settings are necessary, which should be specified in the font's configuration file (see next section).

5.7 Configuration files

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

If you embark 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`; any spaces in the font name should be removed, e.g., `mt-MinionPro.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](#)). It is thus possible to put settings for, e.g., the fonts `pplx` (expert set), `pplj` (oldstyle numerals) and `ppl` (plain) into one and the

Table 3:

Fonts with tailored protrusion settings

Font family (NFSS code)	Features	
<i>Type 1 fonts</i>	Encodings	Shapes
Generic	OT1, T1, T2A, LY1, QX, (TS1) ^a	n, (it, sl, sc) ^a
Computer Modern Roman (cmr) ^b	OT1, OT4, T1, T2A, T5, LY1, TS1	n, it, sl, sc
Bitstream Charter (bch) ^c	OT1, T1, T5, LY1, TS1	n, it, (sl) ^d , sc
EB Garamond ^e	OT1, T1, LY1, TS1	n, it, (sl) ^d , sc
URW Garamond (ugm) ^f	OT1, T1, TS1	n, it
Bitstream Letter Gothic (blg) ^g	OT1, T1, TS1	n, it
Adobe Minion (pmnx, pmnj)	OT1, T1, T2A, 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) ^j	OML/OMS	n/it
AMS symbols (msa, msb)	U	n
Euler (eur, eus, euf) ^k	U	n
Euro symbols (Adobe, ITC, marvosym)	U/OT1	n, it
<i>OpenType fonts</i>	Scripts	Shapes
Generic	Latin	n, (it, sl, sc) ^a
Latin Modern Roman ^l	Latin, Greek	n, it, (sl) ^d
Charis SIL	Latin, Cyrillic, Greek	n, it, sc
Palatino ^m	Latin	n, it, sc

^a Incomplete
^b Aliases: Latin Modern Roman (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor), mmodern (mlmr)
^c Aliases: mathdesign/Charter (mdbch), MicroPress's chmath (chr), XCharter
^d Settings inherited from italic shape
^e Alias: Adobe Garamond (pad, padx, padj)
^f Aliases: mathdesign/URW Garamond (mdugm), garamondx (zgm, zgmj)
^g Alias: ulgothic (ulg)
^h Aliases: pxfonts (pxr), qfonts/QuasiPalatino, T_EX Gyre Pagella (qpl), newpx, FPL Neu (fp9x, fp9j), domitian
ⁱ Aliases: txfonts (txr), qfonts/QuasiTimes, T_EX Gyre Termes (qtm), newtx, tempora, step, stix/stix2
^j Aliases: Latin Modern (lmsy, lmm), mmodern (mlmsy, mlmm)
^k Alias: eulervm (zeur, zeus)
^l Alias: New Computer Modern
^m Aliases: Palatino Linotype, Palatino LT Std, T_EX Gyre Pagella, Domitian

same file `mt-ppl.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,-LF,-TLF,-OsF,-T0sF}
```

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

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

6 Context-sensitive setup

The microtype package also allows applying different micro-typographic settings to the fonts depending on the context in which they occur. 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**, (or **activate** as a shortcut for both), **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 L^AT_EX 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} }
```

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}
```

⁹ Font package authors might also want to have a look at the hook `\Microtype@Hook`, described in the implementation part, section 14.4.4.

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{kerning=french}
```

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

```
\textmicrotypecontext{kerning=}{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}
  {kerning=french, spacing=}
```

Consequently, whenever you switch to the French language, the kerning context will be changed to ‘french’ and the spacing context will be reset. This hook only has an effect if the package 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

pdfTeX 1.40 | LuaTeX 0.62

`\textls` $[\langle amount \rangle] \{\langle general\ text \rangle\}$

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 (independently of whether the tracking option is enabled) in the same way as L^AT_EX’s text commands: `\textls` – which also works in math mode – expects the text in the mandatory argument, while `\lsstyle` will

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

switch on letterspacing for all subsequent fonts until the end of the current group.

`\textls*` 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` `{\langle ligature \rangle}`

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 of solving 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, ‘Ausſichtslosigkeit’, with ligatures shown in green, inhibited ligatures in red).

```
\SetTracking[no ligatures={f}]{encoding = LY, family = yfrak}{120}
\textfrak{\lsstyle Aus{s{\kern0pt}ichts:los{\kern0pt}igkeit}
```

```
\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{120}
\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 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

pdfTeX 1.30 | LuaTeX 0.30

`\DisableLigatures` `[\langle characters \rangle] {\langle set of fonts \rangle}`

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 ?‘ and !‘, but not fi, –, », etc.
```

Only the character that begins the ligature(s) should be specified. This command may only be used in the preamble, and only once.¹¹

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. 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 and LuaTeX, which use a different technique of expansion, the increase of file size can be neglected.

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}
```

¹¹ With LuaTeX, you have to load the fonts with the `fontspec` option ‘`Renderer=Basic`’.

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 activating, say, the font set ‘alltext-nott’). While the `\microtypesetup` command has of course been designed for cases like this, you may find it tiresome to repeat it every time if you are using the `verbatim` environment frequently. The following line (which requires the `etoolbox` package), added to the document’s preamble, would serve the same purpose:

```
\AtBeginEnvironment{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.

Settings for Greek/Thai/Armenian 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., LGR, T2B, 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, 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.

Adjustment of interword spacing is still experimental. The implementation of this feature in `pdfTeX` is not complete, and may not yield the positive effects on the typographical quality you might expect – in certain situations, there may even be undesired side effects, in particular, when used together with the `ragged2e` package. Therefore, the `spacing` option should not be chosen blindly; it is also recommended to experiment with the settings in order to understand the workings of this feature.

Compatibility and interaction with other packages: The `microtype` package is supposed to work happily together with all other `LATEX` packages (except for `pdfcpot`). However, life isn’t perfect, so problems are to be expected. Currently, I am aware of the following issues:

- Even though all configuration files are still provided in legacy (7-bit) format, using multi-byte (Unicode) characters in the settings should run smoothly with an up-to-

date L^AT_EX system. For older systems or documents in legacy encodings, in contrast, this requires loading the `inputenc` package first. Furthermore, 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`.
- Before this package was fully compatible with Lua_T_EX, the following method of enabling expansion and protrusion with the `fontspec` package was most often found to be recommended:

```
\newfontfeature{Microtype}{protrusion=default;expansion=default}
\defaultfontfeatures{Microtype}
```

This code should *not* be used with this package, as it will basically override all of the settings made by `microtype` – despite the naming, the above lines have nothing to do with this package.¹²

- With pdf_T_EX, it is currently not possible to create character-specific settings for Chinese/Japanese/Korean fonts. Therefore, the only micro-typographic extension that can be made to work with CJK fonts is (non-selected) font expansion.
- When used with the `xeCJK` package or the `luatexja` package, text commands (e.g., `\A`, `\textless`) in the configuration will not be understood. You therefore have to ensure that `microtype` will encounter none of them. This requires, firstly, that the glyphs be specified only as single (possibly Unicode) characters, as numbers, or as glyph names (cf. section 5); and secondly, if you are using a font for which pre-defined settings do not exist, that you create these settings yourself (because otherwise, the default settings will be loaded, which do contain text commands). Furthermore, you should load `microtype` late.

Possible error messages and how to get rid of them (specs may differ):

- ! Font csnameendcsname=*cmr10+20 at 10.0pt* not loadable: Metric (TFM) file not found.
This error message will occur if you are trying to employ font expansion while creating DVI output. Remember that *automatic* font expansion only works when running pdf_T_EX or Lua_T_EX in PDF mode. Although expansion is also possible in DVI mode with pdf_T_EX, it requires that all instances of the expanded fonts exist on your _T_EX system.
- ! pdf_T_EX error (font expansion): auto expansion is only possible with scalable fonts.
Automatic font expansion has been improved in pdf_T_EX 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: pdf_L_AT_EX: font *ptmr8r* cannot be expanded (not an included Type1 font)
and the PDF viewer complains about a missing font, e.g., Adobe Reader thusly:
Could not find a font in the Resources dictionary - using Helvetica instead.
With pdf_T_EX 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

¹² They make use of features provided by `luaotfload` (via `fontspec`).

T_EX system is not set up to embed (or ‘download’) the base PostScript fonts (e.g., Times, Helvetica, Courier). In most T_EX distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to true.

- Warning: pdf_latex (file `ecrm1000+20`): Font `ecrm1000+20 at 1200` not found
Furthermore, pdfT_EX 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 pdfT_EX versions, this is only possible if you manually create expanded instances of the fonts.
- ! Font `T1/cmr/m/n/10=ecrm1000 at 10.0pt` not loaded: Not enough room left.
Memory parameter ‘font_mem_size’ too small.
- ! TeX capacity exceeded, sorry [maximum internal font number (font_max)=2000].
Memory parameter ‘font_max’ too small.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf_mem_size)=65536].
Memory parameter ‘pdf_mem_size’ too small (pdfT_EX versions older than 1.30).
When applying micro-typographic enhancement to a large document with a lot of fonts, pdfT_EX 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_EX Live, change the settings in `texmf.cnf`, for MiK_TE_X, in the file `miktex.ini` (2.4 or older) resp. `pdflatex.ini` (2.5 or newer).
- pdfT_EX warning (font expansion): font should be expanded before its first use
This warning will occur with pdfT_EX versions older than 1.40.4, if tracking *and* expansion is applied to a font. It is harmless and can be ignored.

The source code of this document is freely available. If you wonder how this document was created, just have a look at the source code in `microtype.dtx`, which is either already included in your T_EX distribution, or else can be downloaded from [CTAN](#). For the source code of the logo on the title page and of the letterspacing sample from section 5.3, see the appendices A and B. If you want to re-typeset the documentation, read the comments at the end of `microtype.dtx`.

10 Contributions

I would be glad to include configuration files for more fonts. Preparing such configurations is quite a time-consuming task and requires a lot of patience. To alleviate this process, this package also includes a test file that can be used to check at least the protrusion settings (`test-microtype.tex`). If you have created a configuration file for another font, or if you have any suggestions for enhancements in the default configuration files, I would gratefully accept them: w.m.l@gmx.net.

11 Acknowledgments

This package would be pointless if *Hàn Thế Thành* hadn’t created the pdfT_EX programme in the first place, which introduced the micro-typographic extensions and made them available to the T_EX world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004](#), [Thành 2008](#)

and elsewhere. I also thank him and the rest of the pdfTeX team, and more recently also the LuaTeX and XeTeX teams, for refuting the idea that TeX 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. *Georg Duffner* has patiently tested microtype under XeTeX and LuaTeX with his beautiful OpenType font EB Garamond¹³. My thanks also go to *Maciej Eder* for contributing settings for the QX encoding, as well as to *Karl Karlsson* for providing settings for the Cyrillic T2A encoding, and to *Hendrik Vogt*, who made substantial improvements to the Computer Modern Roman italic settings. I thank *Loren B. Davis* for providing protrusion settings for OpenType versions of Palatino Linotype. I am also very much indebted to *Élie Roux*, who not only contributed the lua module in the first place, but also, together with *Philipp Gesang*, took care of updating it for the developments in LuaTeX land.

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. Likewise, *Donald Arseneau* patched his shapepar package to accommodate protrusion.

Additionally, the following people have reported bugs, made suggestions or helped otherwise (in chronological order, quotes indicate TeX.SX user names): *Tom Kink*, *Herb Schulz*, *Michael Hoppe*, *Gary L. Gray*, *Georg Verweyen*, *Christoph Bier*, *Peter Muthesius*, *Bernard Gaille*, *Adam Kucharczyk*, *Mark Rossi*, *Stephan Hennig*, *Michael Zedler*, *Herbert Voß*, *Ralf Stubner*, *Holger Uhr*, *Peter Dyballa*, *Morten Høgholm*, *Steven Bath*, *Daniel Flipo*, *Michalis Miatidis*, *Sven Naumann*, *Ross Hetherington*, *Wiebke Petersen*, *Geoff Vallis*, *Steven E. Harris*, *Karl Berry*, *Peter Meier*, *Nathan Rosenblum*, *Wolfram Schaalo*, *Vasile Gaburici*, *Sveinung Heggen*, *Axel Berger*, *Colin Rourke*, *Maverick Woo*, *Silas S. Brown*, *Lars Rönnbäck*, *Christian Stark*, *Leo*, *Marcin Borkowski*, *hscm*, *George Gratzner*, *Josep Maria Font*, *Juan Acevedo*, *Heiko Oberdiek*, *Till A. Heilmann*, *Rolf Dieterich*, *Seamus Bradley*, *Meho R*, *Steffen Hoffmann*, *Scott Pakin*, *Maïeul Rouquette*, *Jonas Hogstrom*, *Gabriel Kerneis*, ‘RazorXsr’, *Sebastian Schubert*, ‘Dave’, *Giuseppe Palma*, *Stephan Stiller*, *Christopher Schramm*, ‘uli’, *Sam Mason*, ‘kleenstar’, ‘Henning’, *Ronnie Marks*, *David Carlisle*, ‘Max’, ‘HcN’, *Will Robertson*, ‘user11126’, *Ulrike Fischer*, ‘Daniel’, ‘ltcomdata’, *Reinhard Kotucha*, ‘jcr’, *Nils Anders Danielsson*, *Paolo Ney*, *Frank Mittelbach*, *Franz Wexler*, *Moritz Wemheuer*, ‘Andy N’, *Phelype Oleinik*, *Falk Hanisch*, *Markus Kohm*, *Paolo Polesana*, *Oliver Kopp*, *Hironori Kitagawa*, *Daniel Benjamin Miller*, *Md Ayquassar*, *Marcel Krüger*, *Ekkehart Schlicht*, ‘Canageek’, ‘dsedivec’ and ‘DORpapist’.

13 Available from CTAN at pkg/ebgaramond, including configuration files for microtype.

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: ‘EuroBach \TeX 2007 Proceedings’, pp. 36–41. (Online at <http://www.tug.org/TUGboat/Articles/tb29-1/tb91thanh-fonts.pdf>)

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, Martin Schröder, Karl Berry, *The pdf \TeX user manual*, 2 March 2020. (Available from CTAN at [/systems/doc/pdftex/manual/pdftex-a.pdf](#))

Karl Berry, *Fontname: Filenames for \TeX fonts*, July 2009. (Available from CTAN at [/info/fontname/fontname.pdf](#))

\LaTeX 3 Project Team, *\LaTeX 2 ϵ font selection*, November 2020. (Available from CTAN at [/macros/latex/base/fntguide.pdf](#))

Will Robertson, *The fontspec package: Font selection for $X_{\text{\LaTeX}}$ and Lua \LaTeX* , 21 February 2020. (Available from CTAN at [pkg/fontspec](#))

Élie Roux, Khaled Hosny, Philipp Gesang, Ulrike Fischer, Marcel Krüger, *The luaotfload package*, 8 January 2021. (Available from CTAN at [pkg/luaotfload](#))

Carsten Schurig, Tobias Schlemmer, *The pdfcp \TeX .sty package*, 10 June 2005. (Available from CTAN at [pkg/pdfcp \$\text{\TeX}\$](#))

Melchior Franz, *The soul package*, 17 November 2003. (Available from CTAN at [pkg/soul](#)). See also Heiko Oberdiek’s extension of this package, `soulut \TeX 8`, which adds Unicode support. (Available from CTAN at [pkg/soulut \$\text{\TeX}\$ 8](#))

13 Short history

The comprehensive list of changes can be found in appendix C. The following is a list of all changes relevant in the user land; bug and compatibility fixes are swept under the rug. Numbers in brackets indicate the relevant section in this manual.

2.8 (2020/12/07)

- New default font sets for expansion and spacing: ‘all \TeX -nott’ [4, table 2]

2.7 (2017/07/07)

- Allow automatic expansion and letterspacing with Lua \TeX in DVI mode (aka. `dvi \LaTeX`) [3.1, 3.3, table 1]
- Compatibility with \LaTeX 2017/01/01 (fix warnings)

2.6 (2016/05/01)

- Support for Lua \TeX ≥ 0.85
- Improvements for tracking/letterspacing with Lua \TeX (Renderer=Basic no longer required)

- New font sets: ‘alltext-nott’, ‘allmath-nott’ [4, table 2]

2.5 (2013/03/13)

- Support for the `fontspec` package, viz. for OpenType fonts with LuaTeX and XeTeX
- Support for protrusion with XeTeX ≥ 0.9997
- Support for tracking/letterspacing with LuaTeX ≥ 0.62
- Allow context-sensitive setup with LuaTeX
- Info if protrusion settings are generic
- Protrusion settings for Latin Modern Roman (OpenType)
- Protrusion settings for Charis SIL (OpenType)
- Protrusion settings for Palatino Linotype (OpenType)

2.4 (2010/01/10)

- Protrusion settings for T2A encoded Minion

2.3e (2009/11/09)

- Support for the Cyrillic T2A encoding (protrusion, expansion, spacing)

2.3d (2009/03/27)

- New default for expansion option ‘step’: 1, if pdfTeX ≥ 1.40 [3.3]

2.3c (2008/11/11)

- Support for LuaTeX enabled by default

2.3 (2007/12/23)

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

2.2 (2007/07/14)

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

2.1 (2007/01/21)

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

2.0 (2007/01/14)

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

- New commands `\textls` and `\lststyle` for letterspacing, new option ‘`letterspace`’ [3.4, 7]
- New option ‘`babel`’ for automatic micro-typographic adjustment to the selected language [3.5, 6]
- New font sets: ‘`smallcaps`’, ‘`footnotesize`’, ‘`scriptsize`’ [4, table 2]
- New package ‘`letterspace`’ providing the commands for robust and hyphenatable letterspacing [7]

1.9e (2006/07/28)

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

1.9d (2006/05/05)

- Support for the Central European QX encoding (protrusion, inheritance)
- 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
- Inside the preamble, `\microtypesetup` accepts all package options [3.6]
- Protrusion settings for T5 encoded Charter

1.9 (2005/10/28)

- New command `\DisableLigatures` to disable ligatures (pdfTeX ≥ 1.30) [8]
- New command `\microtypecontext` to change the configuration context; new key ‘`context`’ for the configuration commands [6]
- New key ‘`font`’ to add single fonts to the font sets [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 [4]
- New option ‘`config`’ to load a different configuration file [3.5]
- New option ‘`unit`’ to measure protrusion factors relative to a dimension instead of the character width [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
- Support for protrusion with the `ledmac` package (pdfTeX ≥ 1.30)

1.7 (2005/03/23)

- Possibility to specify ranges of font sizes in the set declarations [4, 5]
- New command `\LoadMicrotypeFile` to load a configuration file manually [5.7]
- New command `\Microtype@Hook` for font package authors [14.4.4]
- 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 [3.2, 5]
- When pdfTeX is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled [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 [3.1]
- New option ‘selected’ to enable selected expansion, default: false [3.3, 5.2]
- New default for expansion option ‘step’: 4 (min(stretch,shrink)/5) [3.3]
- Protrusion settings for Bitstream Charter

1.4 (2004/11/12)

- Set up fonts independently from L^AT_EX font loading
- New option: ‘final’ [3.5]

1.2 (2004/10/03)

- New font sets: ‘allmath’ and ‘basicmath’ [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
- New command: \DeclareCharacterInheritance [5.6]
- Characters may also be specified as octal or hexadecimal numbers [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).

pdftex-def: Definitions specific to pdfTeX (microtype-pdftex.def).

xetex-def: Definitions specific to XeTeX (microtype-xetex.def).

luatex-def: Definitions specific to LuaTeX (microtype-luatex.def).

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

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

debug: Code for additional output in the log file.

Used for – surprise! – debugging purposes.

luafile: Lua functions (microtype.lua).

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).

ebg: Settings for EB Garamond (mt-EBGaramond.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).

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 \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 pdfTeX.

```
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*\lsstyle{}
52 \newcommand\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

```

Don't load letterspace.

```

65 \expandafter\let\csname ver@letterspace.sty\endcsname\@empty

```

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

```

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

```

`\MT@warning` Communicate.

```

\MT@warning@nl 76 \def\MT@warning{\PackageWarning\MT@MT{
\MT@info 77 \def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
78 <package>
\MT@info@nl 79 \def\MT@info{\PackageInfo\MT@MT{
\MT@vinfo 80 \def\MT@info@nl#1{\MT@info{#1\@gobble}}
\MT@error 81 \let\MT@vinfo\@gobble
\MT@warn@err 82 \def\MT@error{\PackageError\MT@MT{
83 \def\MT@warn@err#1{\MT@error{#1}{%
84 This error message appears because you loaded the \MT@MT'\MessageBreak
85 package with the option `verbose=errors'. Consult the documentation\MessageBreak
86 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

```

```

87 <debug>
88 \MT@warning@nl{This is the debug version}
89 \newcount\tracingmicrotype

```

```

90 \tracingmicrotype=2
91 \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
92 \def\MT@info@n1#1{\PackageInfo\MT@MT{#1@gobble}\MT@addto@annot{#1}}
93 \let\MT@vinfo\MT@info@n1
94 \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
95 \def\MT@warning@n1#1{\PackageWarning\MT@MT{#1@gobble}\MT@addto@annot{Warning: #1}}
96 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
97 \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

```
98 \newcount\tracingmicrotypeinpdf
```

Let's see how it works ... (if you don't see anything special on this page, your PDF viewer doesn't support annotations).

```
\tracingmicrotypeinpdf=2
```

\MT@pdf@annot
\MT@addto@annot
\ifMT@inannot

During font setup, we save the text for the popup in \MT@pdf@annot. (This requires pdfTeX ≥ 1.30 .) The pdftexcmds package provides pdfTeX's utility commands in LuaTeX, too.

```

99 \RequirePackage{pdftexcmds}
100 \newif\ifMT@inannot \MT@inannottrue
101 \let\MT@pdf@annot\empty
102 \def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
103   {\def\MessageBreak{^J\@spaces}%
104    \MT@xadd\MT@pdf@annot{\pdf@escapestring{#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.

```
105 \newif\iftracingmicrotypeinpdfall
```

\MT@show@pdfannot

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).

```

106 \def\MT@show@pdfannot#1{%
107   \ifnum\tracingmicrotypeinpdf<#1 \else
108     \iftracingmicrotypeinpdfall\leavevmode\fi
109     \pdfannot height 4pt width 4pt depth 2pt {%
110       /Subtype/Caret
111       /T(\expandafter\string\font@name)
112       \ifcase#1\or
113       /Subj(New font)/C[1 0 0]
114       \else
115       /Subj(Known font)/C[0 1 0]
116       \fi
117       /Contents(\MT@pdf@annot)
118     }%
119     \iftracingmicrotypeinpdfall\kern1pt \fi
120     \global\MT@inannotfalse
121   \fi
122 }
123 </debug>
124 </package>

```

14.1.2 Requirements

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

- 0: miniltx
- 1: eplain
- 2: L^AT_EX

For plain usage, we have to copy some commands from `latex.ltx`.

```

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

```

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

```

145 \def\MT@requires@latex#1{%
146   \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
147 }
148 <plain>

```

For definitions that depend on e-TeX features.

```

149 \ifcase 0%
150   \ifx\TeXversion\undefined 1\else
151     \ifx\TeXversion\relax 1\else
152       \ifcase\TeXversion 1\fi
153     \fi
154   \fi
155 \else
156   \catcode`\^^Q=9 \catcode`\^^X=14
157 \fi
158 <letterspace>^^Q\MT@warning@n1{This package requires the etex extensions.
159 <letterspace>^^Q \MessageBreak Exiting}\MT@restore@catcodes\endinput
160 <debug>\MT@info@n1{0}{this is
161 <debug>^^Q not
162 <debug> etex}

```

We check whether we are running pdf_TE_X, X_Y_TE_X, or Lua_TE_X, and load the appropriate definition file.

`\MT@clear@options` If we are using neither of these engines, we disable everything and exit.

```

163 \def\MT@clear@options{%
164 <plain> \MT@requires@latex1{%
165   \AtEndOfPackage{\let\unprocessedoptions\relax\MT@restore@catcodes}%
166   \let\CurrentOption\empty
167 <package> \let\MT@endinput\endinput
168 <plain> }\relax
169 }

```

A hack circumventing the T_EX Live 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_EX Live 2005.

```
170 \ifx\normalpdftexversion\@undefined \else
171   \let\pdftexversion\normalpdftexversion
172   \let\pdftexrevision\normalpdftexrevision
173   \let\pdfoutput\normalpdfoutput
174 \fi
```

\MT@engine Old packages might have let \pdftexversion to \relax.

```
\MT@engine@toold 175 \let\MT@engine\relax
176 <letterspace>\def\MT@engine@toold{0}
177 \ifx\pdftexversion\@undefined \else
178   \ifx\pdftexversion\relax \else
179     \def\MT@engine{pdf}
180 <letterspace>   \let\MT@pdf@or@lua\@firstoftwo
181 <letterspace>   \ifnum\pdftexversion > 139 \def\MT@engine@toold{1}\fi
182 \fi
183 \fi
184 \ifx\directlua\@undefined \else
185   \ifx\directlua\relax \else
186     \def\MT@engine{lua}
```

Since approx. LuaT_EX 0.80, \pdftexversion is let to \luatexversion, so that we would be fooled to think that pdfT_EX is too old.

```
187 <*letterspace>
188   \let\MT@pdf@or@lua\@secondoftwo
189   \ifnum\luatexversion < 62 \def\MT@engine@toold{0}
190   \else
191     \def\MT@engine@toold{1}
192     \let\MT@lua\directlua
193     \ifnum\luatexversion > 84
194       \let\pdfoutput\outputmode
195       \let\pdfprotrudechars\protrudechars
196       \let\pdfadjustspacing\adjustspacing
197     \fi
198   \fi
199 </letterspace>
200 \fi
201 \fi
202 <*package>
203 \ifx\MT@engine\relax
204   \ifx\XeTeXversion\@undefined \else
205     \ifx\XeTeXversion\relax \else
206       \def\MT@engine{xe}
207     \fi
208   \fi
209 \fi
210 </package>
211 </package|letterspace>
```

\MT@pdftex@no pdfT_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 pdfT_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 pdfT_EX:

- 0: not running pdfT_EX
- 1: pdfT_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; scale factor fixed to 1000; default `\efcode = 1000` (≥ 1.20)
- 5: + `\(left,right)marginkern`; `\pdfnoligatures`; `\pdfstrcmp`; `\pdfescapestring` (≥ 1.30)
- 6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`¹⁴; `\pdftracingfonts`; always e-TeX (≥ 1.40)
- 7: + `\letterspacefont` doesn't disable ligatures and kerns; `\pdfcopyfont` ($\geq 1.40.4$)

```

212 (*pdfTeX-def)
213 (debug)\MT@info@nl{0}{this is pdfTeX \the\pdfTeXversion(\pdfTeXrevision)}
214 \def\MT@pdfTeX@no{7}
215 \ifnum\pdfTeXversion = 140
216   \ifnum\pdfTeXrevision < 4
217     \def\MT@pdfTeX@no{6}
218   \fi
219 \else
220   \ifnum\pdfTeXversion < 140
221     \def\MT@pdfTeX@no{5}
222     \ifnum\pdfTeXversion < 130
223       \def\MT@pdfTeX@no{4}
224       \ifnum\pdfTeXversion < 120
225         \def\MT@pdfTeX@no{3}
226         \ifnum\pdfTeXversion = 14
227           \ifnum \expandafter`\pdfTeXrevision < `h
228             \def\MT@pdfTeX@no{2}
229           \ifnum \expandafter`\pdfTeXrevision < `f
230             \def\MT@pdfTeX@no{1}
231           \fi
232         \fi
233       \else
234         \ifnum\pdfTeXversion < 14
235           \def\MT@pdfTeX@no{1}
236         \fi
237       \fi
238     \fi
239   \fi
240 \fi
241 \fi
242 (debug)\MT@info@nl{0}{pdfTeX no.: \MT@pdfTeX@no}
243 (pdfTeX-def)

```

`\MT@xetex@no` XeTeX supports character protrusion since version 0.9997.

```

244 (*xetex-def)
245 (debug)\MT@info@nl{0}{this is xetex (\the\XeTeXversion\XeTeXrevision)}
246 \ifdim 0\XeTeXrevision pt < 0.9997pt
247   \def\MT@xetex@no{1}
248 \else
249   \def\MT@xetex@no{2}
250 \fi
251 (debug)\MT@info@nl{0}{xetex no.: \MT@xetex@no}
252 (xetex-def)

```

`\MT@luaTeX@no` Cases for LuaTeX (`\luaTeXversion` ought to have been enabled by the format):

- 0: N/A
- 1: LuaTeX (< 0.36)
- 2: + `\directlua` without state number (≥ 0.36)

¹⁴ This command was actually introduced in 1.30, but failed on strings longer than 1023 bytes.

- 3: + `\letterspacefont`; non-automatic expansion doesn't work anymore, and automatic expansion in DVI mode is realised by modifying the tracking, not the glyphs¹⁵ (≥ 0.62)
- 4: + almost all of the pdfTeX primitives have been renamed (≥ 0.85)
- 5: + default `\efcode = 1000`; `\protrusionboundary` [not yet supported] (≥ 0.90)
- 6: + `\glet` (≥ 1.10)

Also, sometime between 1.0.4 and 1.0.7, the function `font.setexpansion` has been introduced (but we're not using it for now).

```

253 <lua-def>
254 <debug> \MT@info@nl{this is luatex (\the\luatexversion)}

\MT@lua    Communicate with lua. Beginning with LuaTeX 0.36, \directlua no longer requires
           a state number.

255 \let\MT@lua\directlua
256 \def\MT@luatex@no{6}
257 \ifnum\luatexversion<110
258   \def\MT@luatex@no{5}
259   \ifnum\luatexversion<90
260     \def\MT@luatex@no{4}
261     \ifnum\luatexversion<85
262       \def\MT@luatex@no{3}
263       \ifnum\luatexversion<62
264         \def\MT@luatex@no{2}
265         \ifnum\luatexversion<36
266           \def\MT@lua{\directlua0}
267           \def\MT@luatex@no{1}
268         \fi
269       \fi
270     \fi
271   \fi
272 \fi

273 <debug> \MT@info@nl{0}{luatex no.: \MT@luatex@no}
274 </lua-def>

275 <pdf-def|xetex-def|letterspace>
276 \ifnum
277 <pdf-def|xetex-def> \csname MT@\MT@engine tex@no\endcsname < 2
278 <letterspace> \MT@engine@toold=\z@
279 \MT@warning@nl{You
280 <letterspace>
281   \ifx\MT@engine\relax
282     don't seem to be using pdfTeX or luatex.\MessageBreak
283     Try running 'pdfTeX' or 'luatex' instead of.\MessageBreak
284     \ifx\XeTeXversion\undefined\else xe\fi tex'%
285   \else
286 </letterspace>
287     are using a \MT@engine tex version older than
288 <pdf-def>         0.14f%
289 <xetex-def>       0.9997%
290 <letterspace>     \MT@pdf@or@lua{1.40}{0.62}%
291     .\MessageBreak
292     '\MT@MT' does not work with this version.\MessageBreak
293     Please install a newer version of \MT@engine tex%
294 <letterspace>     \fi
295     .\MessageBreak I will quit now}
296 \MT@clear@options
297 \endinput\fi
298 </pdf-def|xetex-def|letterspace>

```

15 This may have been changed earlier, but I'm no longer able to find out when (the last version that actually works for me is 0.40).

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

```
299 <package>|letterspace>
300 \RequirePackage{keyval}[1997/11/10]
301 <package>
```

`\MT@toks` We need a token register.

```
302 \newtoks\MT@toks
```

`\ifMT@if@` A scratch if.

```
303 \newif\ifMT@if@
```

14.1.3 Declarations

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

```
\ifMT@expansion 304 \newif\ifMT@protrusion
\ifMT@auto       305 \newif\ifMT@expansion
\ifMT@selected   306 \newif\ifMT@auto
\ifMT@noligatures 307 \newif\ifMT@selected
\ifMT@draft       308 \newif\ifMT@noligatures
\ifMT@spacing     309 \newif\ifMT@draft
\ifMT@kerning     310 \newif\ifMT@spacing
\ifMT@tracking    311 \newif\ifMT@kerning
\ifMT@babel       312 \newif\ifMT@tracking
\ifMT@babel       313 \newif\ifMT@babel
```

`\ifMT@babel` [This line intentionally left blank.]

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

```
\MT@ex@level 314 \let\MT@pr@level\tw@
\MT@pr@factor 315 \let\MT@ex@level\tw@
\MT@ex@factor 316 \let\MT@pr@factor\@m
\MT@sp@factor 317 \let\MT@ex@factor\@m
\MT@kn@factor 318 \let\MT@sp@factor\@m
\MT@kn@factor 319 \let\MT@kn@factor\@m
```

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

```
\MT@sp@unit
\MT@kn@unit 320 \let\MT@pr@unit\@empty
321 \let\MT@sp@unit\m@ne
322 \def\MT@kn@unit{1em}
```

`\MT@stretch` Expansion settings.

```
\MT@shrink 323 \let\MT@stretch\m@ne
\MT@step    324 \let\MT@shrink \m@ne
325 \let\MT@step \m@ne
```

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

```
\MT@pr@max 326 \def\MT@pr@min{-\@m}
\MT@ex@min 327 \let\MT@pr@max\@m
\MT@ex@max 328 \let\MT@ex@min\z@
\MT@sp@min 329 \let\MT@ex@max\@m
\MT@sp@max 330 \def\MT@sp@min{-\@m}
\MT@kn@min 331 \let\MT@sp@max\@m
\MT@kn@max 332 \def\MT@kn@min{-\@m}
\MT@tr@min 333 \let\MT@kn@max\@m
\MT@tr@max 334 <package>
335 \def\MT@tr@min{-\@m}
336 \let\MT@tr@max\@m
337 <package>
```

`\MT@factor@default` Default factor.

```
338 \def\MT@factor@default{1000 }
```

```

\MT@stretch@default      Default values for expansion.
\MT@shrink@default      339 \def\MT@stretch@default{20 }
                        340 \def\MT@shrink@default{20 }

\MT@letterspace          Default value for letterspacing (in thousandths of 1 em).
\MT@letterspace@default  341 </package>
                        342 \let\MT@letterspace\m@ne
                        343 \def\MT@letterspace@default{100}
                        344 <*package>

\ifMT@document          Our private test whether we're still in the preamble.
                        345 \newif\ifMT@document
                        346 </package>
                        347 </package|letterspace>

```

14.1.4 Auxiliary macros

```

\MT@requires@pdftex      For definitions that depend on a particular pdfTeX resp. LuaTeX version.
\MT@requires@luatex      348 <*pdftex-def| luatex-def>
                        349 \def
                        350 <pdftex-def> \MT@requires@pdftex%
                        351 <luatex-def> \MT@requires@luatex%
                        352 #1{\ifnum
                        353 <pdftex-def> \MT@pdftex@no
                        354 <luatex-def> \MT@luatex@no
                        355 <#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi>
                        356 <luatex-def&debug>\MT@requires@luatex4{\MT@lua{tex.enableprimitives('pdf',{'tracingfonts'})}}\relax
                        357 <pdftex-def&debug>\MT@requires@pdftex6{
                        358 <debug>\pdftracingfonts=1
                        359 <pdftex-def&debug>}\relax
                        360 </pdftex-def| luatex-def>

```

Some functions are loaded from a dedicated lua file. This avoids character escaping problems and incompatibilities between versions of LuaTeX. Unless running a recent L^AT_EX, we load the luatexbase package.

```

361 <*luatex-def>
362 \ifl@t@r\fmtversion{2016/01/01}\relax{\RequirePackage{luatexbase}}

```

We load luaotfload, because some of its functions are required in microtype.lua. This eliminates the need for the user to load fontspec before microtype. There will hardly be any LuaTeX documents that don't load this package, anyway. Since 2017/01/01, it is already loaded in the format.

```

363 \ifl@t@r\fmtversion{2017/01/01}\relax{\RequirePackage{luaotfload}}
364 \MT@lua{require("microtype")}
365 </luatex-def>

```

Here it begins. The module was contributed by Élie Roux.

```

366 <*luafile>
367
368 function microtype.info(...)
369   luatexbase.module_info("microtype",...)
370 end
371
372 local find      = string.find
373 local match     = string.match
374 local tex_write = tex.write
375
376 local catpackage
377 if luatexbase.registernumber then
378   catpackage = luatexbase.registernumber("catcodetable@atletter") -- LaTeX
379 else

```

```

380 catpackage = luatexbase.catcodetables.CatcodeTableLaTeXAtLetter -- luatexbase
381 end
382 function microtype.sprint (...)
383   tex.sprint(catpackage, ...)
384 end
385
386 /luafile

```

To be continued, but first back to primitives.

`\MT@glet` Here's the forgotten one (finally implemented in LuaTeX).

```

387 <luatex-def> \MT@requires@luatex6{\let\MT@glet\glet}\relax
388 <*package|letterspace>
389 \def\MT@glet{\global\let}

```

`\MT@exp@cs` Commands to create command sequences. Those that are going to be defined globally should be created inside a group so that the save stack won't explode.

```

\MT@exp@gcs
390 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
391 <*package>
392 \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 393 \def\MT@def@n{\MT@exp@cs\def}
394 \def\MT@gdef@n{\MT@exp@gcs\gdef}

```

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

```

\MT@xdef@n 395 </package>
396 \def\MT@edef@n{\MT@exp@cs\edef}
397 <*package>
398 \def\MT@xdef@n{\MT@exp@gcs\xdef}

```

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

```

\MT@glet@nc 399 \def\MT@let@nc{\MT@exp@cs\let}
400 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}

```

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

```

401 </package>
402 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}
403 <*package>

```

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

```

\MT@glet@nn 404 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}
405 \def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}

```

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

```

406 \def\MT@@font{\expandafter\string\MT@font}

```

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

```

407 </package>
408 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}

```

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

```

409 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
410 <*package>

```

`\MT@exp@two@n` Expand the next two tokens after `<#1>` once and enclose them in braces.

```

411 \def\MT@exp@two@n#1#2#3{%
412   \expandafter\expandafter\expandafter
413   #1\expandafter\expandafter\expandafter
414   {\expandafter#2\expandafter}\expandafter{#3}}

```

You do not wonder why `\MT@exp@one@c` doesn't exist, do you?

<code>\MT@ifdefined@c@T</code>	Wrapper for testing whether command resp. <code>\csname</code> sequence is defined. If we
<code>\MT@ifdefined@c@TF</code>	are running e-TeX, we will use its primitives <code>\ifdefined</code> and <code>\ifcsname</code> , which
<code>\MT@ifdefined@n@T</code>	decreases memory use substantially.
<code>\MT@ifdefined@n@TF</code>	<pre> 415 \def\MT@ifdefined@c@T#1{% 416 ^^X \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi 417 ^^Q \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi 418 } 419 </package> 420 \def\MT@ifdefined@c@TF#1{% 421 ^^X \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi 422 <package>^^Q \ifx#1\@undefined 423 <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi 424 } 425 \def\MT@ifdefined@n@T#1{% 426 ^^X \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi 427 <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax 428 <package>^^Q \expandafter\@gobble\else\expandafter\@firstofone\fi 429 } 430 \def\MT@ifdefined@n@TF#1{% 431 ^^X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi 432 <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax 433 <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi 434 } 435 <*package> </pre>
<code>\MT@detokenize@n</code>	Translate a macro into a token list. With e-TeX, we can use <code>\detokenize</code> . We also
<code>\MT@detokenize@c</code>	need to remove the last trailing space; and only the last one – therefore the fiddling
<code>\MT@rem@last@space</code>	(and the <code>\string</code> isn't perfect, of course).
	<pre> 436 \def\MT@detokenize@n#1{% 437 ^^X \expandafter\MT@rem@last@space\detokenize{#1} \@nil 438 ^^Q \string#1% 439 } 440 \def\MT@detokenize@c#1{% 441 ^^X \MT@exp@one@n\MT@detokenize@n#1% 442 ^^Q \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil 443 } 444 \def\MT@rem@last@space#1 #2{#1% 445 \ifx\@nil#2\else \space 446 \expandafter\MT@rem@last@space\expandafter#2\fi 447 } </pre>
<code>\MT@ifempty</code>	Test whether argument is empty.
	<pre> 448 </package> 449 \begingroup 450 \catcode`\%=12 451 \catcode`\&=14 452 \gdef\MT@ifempty#1{% 453 \if %#1%& 454 \expandafter\@firstoftwo 455 \else 456 \expandafter\@secondoftwo 457 \fi 458 } 459 \endgroup 460 <*package> </pre>
<code>\MT@ifint</code>	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 <code>letterspace</code> option).
	<pre> 461 </package> 462 </package letterspace> 463 <pdfTeX-def>\MT@requires@pdftex6{ 464 <letterspace>\MT@pdf@or@lua{ </pre>

```

465 <pdfTeX-def|letterspace>
466 \def\MT@ifint#1{%
467   \ifcase\pdfmatch{^-[0-9]+ *$}{#1}\relax
468   \expandafter\@secondoftwo
469   \else
470   \expandafter\@firstoftwo
471   \fi
472 }
473 }{
474 </pdfTeX-def|letterspace>
475 <pdfTeX-def|xetex-def|letterspace>
476 \def\MT@ifint#1{%
477   \if!\ifnum9<1#1!\else?\fi
478   \expandafter\@firstoftwo
479   \else
480   \expandafter\@secondoftwo
481   \fi
482 }
483 </pdfTeX-def|xetex-def|letterspace>
484 <pdfTeX-def|letterspace>}
485 <luatex-def>\def\MT@ifint#1{\csname\MT@lua{microtype.if_int}([#1])\endcsname}
486 <luafile>
487 local function if_int(s)
488   if find(s,"^-[0-9]+ *$") then
489     tex_write("@firstoftwo")
490   else
491     tex_write("@secondoftwo")
492   end
493 end
494 microtype.if_int = if_int
495
496 </luafile>

```

`\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.)

```

497 <pdfTeX-def>
498 \MT@requires@pdfTeX6{
499 \def\MT@ifdimen#1{%
500   \ifcase\pdfmatch{^[0-9]+([.][0-9]+)?|.[0-9]+}%
501   (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
502   \expandafter\@secondoftwo
503   \else
504   \expandafter\@firstoftwo
505   \fi
506 }
507 }{
508 </pdfTeX-def>
509 <pdfTeX-def|xetex-def>
510 \def\MT@ifdimen#1{%
511   \setbox\z@=\hbox{%
512     \MT@count=1#1\relax
513     \ifnum\MT@count=\@ne
514       \aftergroup\@secondoftwo
515     \else
516       \aftergroup\@firstoftwo
517     \fi
518   }%
519 }
520 </pdfTeX-def|xetex-def>
521 <pdfTeX-def>}
522 <luatex-def>\def\MT@ifdimen#1{\csname\MT@lua{microtype.if_dimen}([#1])\endcsname}
523 <luafile>
524 local function if_dimen(s)
525   if (find(s,"^-[0-9]+(%a*) *$") or
526       find(s,"^-[0-9]*[.][0-9]+(%a*) *$")) then

```

```

527     tex_write("@firstoftwo")
528   else
529     tex_write("@secondoftwo")
530   end
531 end
532 microtype.if_dimen = if_dimen
533
534 </luafile>

\MT@ifdim    Compare floating point numbers.

535 <*package>
536 \def\MT@ifdim#1#2#3{%
537   \ifdim #1\p@ #2 #3\p@
538     \expandafter\@firstoftwo
539   \else
540     \expandafter\@secondoftwo
541   \fi
542 }
543 </package>

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

544 <pdfTex-def|XeTeX-def>
545 <pdfTex-def>\MT@requires@pdfTex5{
546 \def\MT@ifstreq#1#2{%
547   <pdfTex-def> \ifnum\pdfstrcmp{#1}{#2}=\z@
548   <XeTeX-def> \ifnum\strcmp{#1}{#2}=\z@
549     \expandafter\@firstoftwo
550   \else
551     \expandafter\@secondoftwo
552   \fi
553 }
554 </pdfTex-def|XeTeX-def>
555 <*pdfTex-def>
556 }{
557 \def\MT@ifstreq#1#2{%
558   \edef\MT@res@a{#1}%
559   \edef\MT@res@b{#2}%
560   \ifx\MT@res@a\MT@res@b
561     \expandafter\@firstoftwo
562   \else
563     \expandafter\@secondoftwo
564   \fi
565 }
566 }
567 </pdfTex-def>
568 <LaTeX-def>\def\MT@ifstreq#1#2{\csname\MT@lua{microtype.if_str_eq}([#1],[#2])\endcsname}
569 <*luafile>
570 local function if_str_eq(s1, s2)
571   if s1 == s2 then
572     tex_write("@firstoftwo")
573   else
574     tex_write("@secondoftwo")
575   end
576 end
577 microtype.if_str_eq = if_str_eq
578
579 </luafile>

\MT@xadd    Add item to a list.

580 <*package>
581 \def\MT@xadd#1#2{%
582   \ifx#1\relax
583     \xdef#1{#2}%
584   \else
585     \xdef#1{#1#2}%

```

```

586 \fi
587 }

\MT@xaddb      Add item to the beginning.
588 \def\MT@xaddb#1#2{%
589 \ifx#1\relax
590 \xdef#1{#2}%
591 \else
592 \xdef#1{#2#1}%
593 \fi
594 }
595 <package>

\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@      596 <package>|letterspace
\MT@clist@function  597 \def\MT@map@clist@n#1#2{%
\MT@clist@break      598 \ifx\@empty#1\else
599 \def\MT@clist@function##1{#2}%
600 \MT@map@clist@#1,\@nil,\@nnil
601 \fi
602 }

603 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
604 \def\MT@map@clist@c#1,%
605 \ifx\@nil#1%
606 \expandafter\MT@clist@break
607 \fi
608 \MT@clist@function{#1}%
609 \MT@map@clist@
610 }
611 \let\MT@clist@function\@gobble
612 \def\MT@clist@break#1\@nnil{}
613 <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@      614 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break      615 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
616 \def\MT@map@tlist@#1#2{%
617 \ifx\@nnil#2\else
618 #1{#2}%
619 \expandafter\MT@map@tlist@
620 \expandafter#1%
621 \fi
622 }
623 \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      624 \newif\ifMT@inlist@
625 \def\MT@in@clist#1#2{%
626 \def\MT@res@a#1,#1,##2##3\@nnil{%
627 \ifx##2\@empty
628 \MT@inlist@false
629 \else
630 \MT@inlist@true
631 \fi
632 }%
633 \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
634 }

\MT@rem@from@clist      Remove item <#1> from comma list <#2>. This is basically \@removeelement from
ltnctr1.dtx. Using \pdfmatch and \pdflastmatch here would be really slow!
635 \def\MT@rem@from@clist#1#2{%

```



```

636 \def\MT@res@a##1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
637 \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
638 \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
639 }

\MT@in@tlist    Test whether item is in token list. Since this isn't too elegant, I thought that at least
\MT@in@tlist@    here, \pdfmatch would be more efficient – however, it turned out to be even slower
                  than this solution.

640 \def\MT@in@tlist#1#2{%
641   \MT@inlist@false
642   \def\MT@res@a{#1}%
643   \MT@map@tlist@c#2\MT@in@tlist@
644 }
645 \def\MT@in@tlist@#1{%
646   \edef\MT@res@b{#1}%
647   \ifx\MT@res@a\MT@res@b
648     \MT@inlist@true
649     \expandafter\MT@tlist@break
650   \fi
651 }

\MT@in@rlist    Test whether size \MT@size is in a list of ranges. Store the name of the list in
\MT@in@rlist@    \MT@size@name
\MT@in@rlist@@ 652 \def\MT@in@rlist#1{%
\MT@size@name 653   \MT@inlist@false
654   \MT@map@tlist@c#1\MT@in@rlist@
655 }
656 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
657 \def\MT@in@rlist@@#1#2#3{%
658   \MT@ifdim{#2}=\m@ne{%
659     \MT@ifdim{#1}=\MT@size
660     \MT@inlist@true
661     \relax
662   }{%
663     \MT@ifdim\MT@size<{#1}\relax{%
664       \MT@ifdim\MT@size<{#2}%
665       \MT@inlist@true
666       \relax
667     }%
668   }%
669   \ifMT@inlist@
670     \def\MT@size@name{#3}%
671     \expandafter\MT@tlist@break
672   \fi
673 }

\MT@loop    This is the same as LATEX's \loop, which we mustn't use, since this could confuse an
\MT@iterate  outer \loop in the document.
\MT@repeat  674 </package>
675 \def\MT@loop#1\MT@repeat{%
676   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
677   \MT@iterate \let\MT@iterate\relax
678 }
679 \let\MT@repeat\fi

\MT@while@num    Execute <#3> from <#1> up to (excluding) <#2> (much faster than LATEX's \@whilenum).
680 \def\MT@while@num#1#2#3{%
681   \@tempcnta#1\relax
682   \MT@loop #3%
683   \advance\@tempcnta \@ne
684   \ifnum\@tempcnta < #2\MT@repeat
685 }
686 </package>|letterspace

```

`\MT@if@luaotf@font` For fonts loaded by `luaotfload` we query the font's table.

```

687 <letterspace>\MT@pdf@or@lua{\let\MT@if@luaotf@font\@secondoftwo}{
688 <luatex-def|letterspace>\def\MT@if@luaotf@font{\csname\MT@lua{%
689 <luatex-def> microtype.if_luaotf_font()
690 <*luafile|letterspace>
691 <luafile>local function if_luaotf_font()
692   local thefont = font.getfont(font.current())
693   if thefont and ( thefont.format == "opentype" or thefont.format == "truetype" )
694     then tex.write("@firstoftwo")
695     else tex.write("@secondoftwo")
696   end
697 <luafile>end
698 <luafile>microtype.if_luaotf_font = if_luaotf_font
699 <luafile>
700 </luafile|letterspace>
701 <luatex-def|letterspace> }\endcsname
702 <luatex-def|letterspace>}
703 <letterspace>}

```

`\MT@do@font` Execute `<#1>` 256 times,

```

704 <pdfTeX-def|letterspace>\def\MT@do@font{\MT@while@num\z@\cc@lvi}

```

resp. for the whole font for LuaTeX, if it's a Unicode font.

```

705 <*luatex-def>
706 \def\MT@do@font#1{%
707   \MT@if@luaotf@font{%
708     \def\MT@dofont@function#1{%
709       \MT@lua{microtype.do_font()}%
710     }{\MT@while@num\z@\cc@lvi{#1}}%
711   }
712 </luatex-def>

```

This is the `lua` function, which is much faster than looping through all glyphs in TeX. Legacy fonts (which this function should never work on) don't contain a `v.index` field.

```

713 <*luafile>
714 local function do_font()
715   local thefont = font.getfont(font.current())
716   if thefont then
717     for i,v in next,thefont.characters do
718       if v.index == nil or v.index > 0 then
719         microtype.sprint([[ \@tempcnta=]]..i..[[\relax\MT@dofont@function]])
720       end
721     end
722   end
723 end
724 microtype.do_font = do_font
725
726 </luafile>

```

The \XeTeX variant (it's slow ...!).

```

727 <*xetex-def>
728 \def\MT@do@font#1{%
729   \@tempcnta=\z@
730   \MT@loop
731     \iffontchar\MT@font\@tempcnta #1\fi
732     \advance\@tempcnta\@ne
733     \ifnum\@tempcnta < \XeTeXlastfontchar\MT@font \MT@repeat
734   }
735 </xetex-def>
736 <*package>

```

`\MT@count` Increment macro `<#1>` by one. Saves using up too many counters. The e-TeX way is `\MT@increment`

slightly faster.

```

737 \newcount\MT@count
738 \def\MT@increment#1{%
739   ^^X \edef#1{\number\numexpr #1 + 1\relax}%
740   ^^Q \MT@count=#1\relax
741   ^^Q \advance\MT@count \@ne
742   ^^Q \edef#1{\number\MT@count}%
743 }
```

`\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.

```

744 \def\MT@scale#1#2#3{%
745   ^^Q \multiply #1 #2\relax
746   \ifnum #3 = \z@
747     ^^X #1=\numexpr #1 * #2\relax
748   \else
749     ^^X #1=\numexpr #1 * #2 / #3\relax
750   ^^Q \divide #1 #3\relax
751   \fi
752 }
```

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

```

\MT@abbr@ex
\MT@abbr@pr@c 753 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 754 \def\MT@abbr@ex{expansion}
755 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@pr@inh 756 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh 757 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@n 758 \def\MT@abbr@ex@inh{expansion inheritance}
759 \def\MT@abbr@n{noligatures}
\MT@abbr@sp 760 \def\MT@abbr@sp{spacing}
\MT@abbr@sp@c 761 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@sp@inh 762 \def\MT@abbr@sp@inh{interword spacing inheritance}
763 \def\MT@abbr@kn{kerning}
\MT@abbr@kn 764 \def\MT@abbr@kn@c{kerning codes}
\MT@abbr@kn@c 765 \def\MT@abbr@kn@inh{kerning inheritance}
766 \def\MT@abbr@tr{tracking}
\MT@abbr@kn@inh 767 \def\MT@abbr@tr@c{tracking amount}
```

`\MT@abbr@tr` These we also need the other way round.

```

\MT@rbba@protrusion
\MT@rbba@tr@c
\MT@rbba@expansion 768 \def\MT@rbba@protrusion{pr}
769 \def\MT@rbba@expansion{ex}
\MT@rbba@spacing 770 \def\MT@rbba@spacing{sp}
\MT@rbba@kerning 771 \def\MT@rbba@kerning{kn}
\MT@rbba@tracking 772 \def\MT@rbba@tracking{tr}
```

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

```

\MT@features@long 773 \def\MT@features{pr,ex,sp,kn,tr}
774 \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 command to check whether a feature exists in order to prevent a rather confusing 'Missing \endcsname inserted' error message. The feature (long form) must be in `<#1>`, the type of list to ignore in `<#2>`, then comes the action.

```

775 \def\MT@is@feature#1#2{%
776   \MT@in@clist{#1}\MT@features@long
777   \ifMT@inlist@
778     \expandafter\@firstofone
779   \else
780     \MT@error{`#1' is not an available micro-typographic\MessageBreak
```

```

781     feature. Ignoring #2}{Available features are: `MT@features@long'.}%
782     \expandafter\@gobble
783   \fi
784 }

```

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)
- `\showhyphens` (in section 14.4.6)

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.

```

785 \@ifl@aded{tex}{wordcount}{%
786   \MT@warning@nl{Detected the `wordcount' utility.\MessageBreak
787     Disabling `MT@MT', since it wouldn't work}%
788   \MT@clear@options\endinput}\relax

```

The minimal class doesn't define any size commands other than `\normal size`, which will result in lots of warnings. Therefore we issue a warning about the warnings.

```

789 \@ifclassloaded{minimal}{%
790   \MT@warning@nl{Detected the `minimal' class.\MessageBreak
791     Expect lots of warnings and some malfunctions.\MessageBreak
792     You might want to use a proper class instead}%
793 }\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.

```

794 </package>
795 <*package|letterspace>
796 <plain>\MT@requires@latex1{
797 \let\MT@setup@{}

```

`\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.

```

798 \def\MT@addto@setup{\g@addto@macro\MT@setup@

```

Don't hesitate with `miniltx`.

```

799 <plain>}{\let\MT@addto@setup\@firstofone}

```

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

```

800 \def\MT@with@package@T#1{\@ifpackageloaded{#1}\@firstofone\@gobble}
801 </package|letterspace>
802 <*package>

```

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

```

803 \def\MT@with@babel@and@T#1{%
804   \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
805     \expandtwoargs\MT@in@clist{#1}
806     {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
807     \ifMT@in@list\expandafter\@gobble\fi
808   }\@gobble
809 }

```

`\MT@ledmac@setup`

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

Peter Wilson was so kind to provide the `\l@dunhbox@line` hook in `ledmac` to allow for protrusion. `\leftmarginkern` and `\rightmarginkern` are new primitives of pdfTeX 1.21b (aka. 1.30.0). They are also part of recent XeTeX. The successor packages `eledmac` and `reledmac` are also supported.

```

810 </package>
811 <pdfTeX-def>\MT@requires@pdfTeX5{
812 <*pdfTeX-def|luaTeX-def|xetex-def>
813   \def\MT@ledmac@setup{%
814     \ifMT@protrusion
815       \MT@ifdefined@c@TF\l@dunhbox@line{%

```

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

```

816       \MT@info@nl{Patching ((r)e)ledmac to enable character protrusion}%
817       \let\MT@led@unhbox@line\l@dunhbox@line
818       \renewcommand*{\l@dunhbox@line}[1]{%
819         \ifhbox##1%
820           \kern\leftmarginkern##1%
821           \expandafter\MT@led@unhbox@line\expandafter##1\expandafter
822           \kern\rightmarginkern##1%
823         \fi
824       }%
825     }{%
826       \MT@warning@nl{%
827         Character protrusion in paragraphs with line\MessageBreak
828         numbering will only work if you update ledmac,\MessageBreak
829         or use one of its successors, eledmac or reledmac}%
830     }%
831   \fi
832 }
833 </pdfTeX-def|luaTeX-def|xetex-def>
834 <*pdfTeX-def>
835 }{
836   \def\MT@ledmac@setup{%
837     \ifMT@protrusion
838       \MT@warning@nl{%
839         The pdfTeX version you are using does not allow\MessageBreak
840         character protrusion in paragraphs with line\MessageBreak
841         numbering by the `((r)e)ledmac' package.\MessageBreak
842         Upgrade pdfTeX to version 1.30 or later}%
843     \fi
844   }
845 }
846 </pdfTeX-def>

```

The `shapepar` package (v2.2) fixes this in a similar manner by itself, so we don't have to bother.

`\MT@restore@p@h`

Restore meaning of `\%` and `\#`.

```

847 <*package|letterspace>
848 <*package>
849 \def\MT@restore@p@h{\chardef\%~\% \chardef\#~\# }

```

`\ifMT@fontspec`

Two new conditionals for use with XeTeX or LuaTeX.

`\ifMT@xunicode`

```

850 \newif\ifMT@fontspec
851 \MT@with@package@T{fontspec}\MT@fontspectrue

```

```

852 \newif\ifMT@xunicode
853 \MT@with@package@T{xunicode}\MT@xunicodetrue

```

We need the correct value of the former for configuration commands inside the preamble (to get the default families right).

```

854 \ifl@t@r\fmtversion{2020/10/01}
855 {\AddToHook{package/after/fontspect}{\MT@fontspectrue}}\relax

```

```

\MT@maybe@gobble@with@tikz
\MT@tikz@setup

```

If `\tikz@expandcount` is greater than zero, we're inside or at the end of a `tikz` node, where we don't want to adjust spacing after letterspacing, lest we disturb `tikz`. This is used in `\MT@afteraftergroup`, and we don't need it for `letterspace`.

```

856 \let\MT@maybe@gobble@with@tikz\@firstofone
857 \def\MT@tikz@setup{%
858   \def\MT@maybe@gobble@with@tikz{%
859     \ifnum\tikz@expandcount>\z@
860       \expandafter\@gobble
861     \else
862       \expandafter\@firstofone
863     \fi}}

```

```
\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`.)

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

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

```

865 \MT@if@false
866 \MT@with@babel@and@T{spanish}\MT@if@true
867 \MT@with@babel@and@T{galician}\MT@if@true
868 \MT@with@babel@and@T{mexican}\MT@if@true
869 \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.)

```

870 \MT@with@package@T{csquotes}{%
871   \@ifpackagelater{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` and `mathastext`.

```

872 \MT@if@false
873 \MT@with@package@T{hyperref}\MT@if@true
874 \MT@with@package@T{tex4ht}\MT@if@true
875 \MT@with@package@T{mathastext}\MT@if@true
876 \ifMT@if@MT@restore@p@h\fi
877 \MT@with@package@T{tikz}\MT@tikz@setup
878 }

```

Check again at the end of the preamble.

```

879 </package>
880 \MT@addto@setup{%
881 <*/package>

```

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

```

882 \MT@with@package@T{pdfcpot}{%
883 \MT@error{Detected the `pdfcpot' package!\MessageBreak
884 \MT@MT' and `pdfcpot' may not be used together}{%
885 The `pdfcpot' package provides an interface to character protrusion.\MessageBreak
886 So does the `MT@MT' package. Using both packages at the same\MessageBreak
887 time will almost certainly lead to undesired results. Have your choice!}%
888 }%
889 \MT@with@package@T {ledmac}\MT@ledmac@setup
890 \MT@with@package@T {eledmac}\MT@ledmac@setup
891 \MT@with@package@T{reledmac}\MT@ledmac@setup
892 \MT@with@package@T{xunicode}\MT@xunicodetrue
893 \MT@with@package@T{fontspec}\MT@fontspectrue

```

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

```

894 \MT@gllet\MT@setupfont@hook\@empty
895 \MT@if@false
896 \MT@with@babel@and@T{spanish} \MT@if@true
897 \MT@with@babel@and@T{galician}\MT@if@true
898 \MT@with@babel@and@T{mexican} \MT@if@true
899 \ifMT@if@
900 \g@addto@macro\MT@setupfont@hook{%
901 \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
902 \fi
903 \MT@with@package@T{csquotes}{%
904 \ifpackage@later{csquotes}{2005/05/11}{%
905 \g@addto@macro\MT@setupfont@hook\@disablequotes
906 }{%
907 \MT@warning@n!{%
908 Should you receive warnings about unknown slot\MessageBreak
909 numbers, try upgrading the `csquotes' package}%
910 }%
911 }%

```

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

```

912 \MT@if@false
913 </package>
914 <plain> \MT@requires@latex2{
915 \MT@with@package@T{hyperref}{%
916 \pdfstringdefDisableCommands{%
917 <*package>
918 \MT@ltx@pickupfont
919 \let\textmicrotypecontext\@secondoftwo
920 \let\microtypecontext\@gobble
921 </package>
922 \def\lsstyle{\pdfstringdefWarn\lsstyle}%
923 \def\textls#1#{\pdfstringdefWarn\textls}%
924 }%
925 <package> \MT@if@true
926 }%
927 <plain> }\relax
928 <*package>
929 \MT@with@package@T{tex4ht}\MT@if@true
930 \MT@with@package@T{mathastext}\MT@if@true
931 \ifMT@if@g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi

```

The listings package makes numbers and letters active,

```

932 \MT@with@package@T{listings}{%
933 \g@addto@macro\MT@cfg@catcodes{%
934 \MT@while@num{"30}{ "3A}{\catcode\@tempcnta=12\relax}%
935 \MT@while@num{"41}{ "5B}{\catcode\@tempcnta=11\relax}%
936 \MT@while@num{"61}{ "7B}{\catcode\@tempcnta=11\relax}%
937 }%

```

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

```
938 \g@addto@macro\MT@setupfont@hook{%
939 \catcode`\=\z@
```

Inside a listing, `\space` is redefined.

```
940 \def\space{ }%
```

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.

```
941 \let\lst@ProcessLetter\empty
942 }%
943 }%
```

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. Also, we have to disable expansion within `soul`'s trial run. Under plain $\text{T}_{\text{E}}\text{X}$, `soul` doesn't register itself the $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ way, so we just test for its main command.

```
944 </package>
945 \ifx\Soul@\undefined\else
946 \soulregister\lsstyle 0%
947 \soulregister\textls 1%
948 \ifx\XeTeXrevision\undefined
949 \let\MT@Soul@doword\Soul@doword
950 \def\Soul@doword{\pdfadjustspacing=\z@ \MT@Soul@doword}%
951 \fi
952 \fi
953 <*package>
954 \MT@with@package@T{tikz}\MT@tikz@setup
```

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` (pre-4.6.0), `\py@macron` had only one argument.

```
955 \MT@with@package@T{pinyin}{%
956 \let\MT@orig@py@macron\py@macron
957 \ifpackagelater{pinyin}{2005/08/11}{% 4.6.0
958 \def\py@macron#1#2{%
959 \MT@ltx@pickupfont
960 \MT@orig@py@macron{#1}{#2}%
961 \MT@MT@pickupfont}%
962 }{%
963 \def\py@macron#1{%
964 \MT@ltx@pickupfont
965 \MT@orig@py@macron{#1}%
966 \MT@MT@pickupfont}%
967 }%
968 }%
969 </package>
970 }
971 </package|letterspace>
```

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

```
972 <package>\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`).

```
973 <*pdfTeX-def|xetex-def|luatex-def>
974 \def\MT@setupfont{%
```


With X_YTeX and LuaTeX the font may not be actually loaded, hence we might see a wrong font (in `\MT@get@slot`). Therefore, we first load the current font.

```
975 <xetex-def|luatex-def> \MT@font
```

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

```
976 \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).

```
977 <pdfTeX-def>\MT@requires@pdfTeX7{
```

```
978 <pdfTeX-def|luatex-def>\g@addto@macro\MT@setupfont\MT@copy@font
```

```
979 <pdfTeX-def>}\relax
```

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

```
980 \g@addto@macro\MT@setupfont{%
```

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

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

```
982 \MT@exp@one@n\MT@find@file\MT@family
```

```
983 \ifx\MT@familyalias\@empty \else
```

```
984 \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.)

```
985 % \ifx\f@encoding\cf@encoding\else\@enc@update\fi
```

```
986 }
```

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

```
987 <pdfTeX-def>\MT@requires@pdfTeX6
```

```
988 <luatex-def>\MT@requires@luatex3
```

```
989 <pdfTeX-def|luatex-def> {\g@addto@macro\MT@setupfont\MT@tracking}\relax
```

```
990 \g@addto@macro\MT@setupfont{%
```

```
991 \MT@check@font
```

```
992 \ifMT@inlist@
```

```
993 <debug>\MT@show@pdfannot2%
```

```
994 \else
```

```
995 \MT@vinfo{Setting up font `~\MT@font'\on@line}%
```

```
996 \MT@info@nottracking
```

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.

```
997 \MT@protrusion
```

```
998 <pdfTeX-def|luatex-def> \MT@expansion
```

```
999 }
```

Interword spacing and kerning (pdfTeX 1.40).

```
1000 <*pdfTeX-def>
```

```
1001 \MT@requires@pdfTeX6{
```

```
1002 \g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}
```

```
1003 }\relax
```

```
1004 </pdfTeX-def>
```

Disable ligatures (pdfTeX 1.30).

```
1005 <pdfTeX-def>\MT@requires@pdfTeX5{
```

```

1006 <pdftex-def>|luatex-def>\g@addto@macro\MT@setupfont\MT@noligatures
1007 <pdftex-def>}\relax
1008 \g@addto@macro\MT@setupfont{%

```

Debugging.

```

1009 <debug>\MT@show@pdfannot1%

```

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

```

1010 \MT@register@font
1011 \fi
1012 }
1013 </pdftex-def>|xtex-def>|luatex-def>

```

\MT@copy@font The new (1.40.4) \pdfcopyfont command allows expanding 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.

```

1014 <*pdftex-def>|luatex-def>
1015 \let\MT@copy@font\relax
1016 <pdftex-def>\MT@requires@pdftex7{
1017 \def\MT@copy@font{%

```

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

```

1018 \xdef\MT@font@copy{\csname\MT@font/\MT@pr@context/\MT@ex@context\endcsname}%
1019 \expandafter\ifx\MT@font@copy\relax

```

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

```

1020 \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
1021 \expandafter\ifx\MT@font@orig\relax
1022 \MT@exp@two@c\MT@gl@et\MT@font@orig\font@name
1023 \else
1024 \MT@exp@two@c\let\font@name\MT@font@orig
1025 \fi
1026 <pdftex-def> \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name

```

Even though LuaTeX also provides the primitive from pdfTeX (even renamed to \copyfont, that is, 'promoted' as per the LuaTeX manual), it is seriously crippled in that OpenType features will be lost. Therefore, we do not copy the font but load it anew.

```

1027 <luatex-def> \MT@exp@two@c\MT@lua@copyfont\meaning\font@name\@nil
1028 <debug>\MT@dinfol{creating new copy: \MT@font@copy}%

```

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

```

1029 \MT@map@clist@c\MT@active@features{%
1030 \MT@exp@cs\ifx\MT@nameuse\MT@abbr@##1}\relax\else
1031 \def\@tempa{##1}%
1032 \MT@exp@cs\MT@map@tlist@c\MT@##1@doc@contexts\MT@rem@from@list
1033 \fi
1034 }%
1035 \fi
1036 \MT@exp@two@c\let\MT@font\MT@font@copy

```

We only need the font identifier for letterspacing.

```

1037 \let\font@name\MT@font@copy

```

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

```

1038 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
1039 }

```

\MT@rem@from@list

```

1040 \def\MT@rem@from@list#1{%
1041   \MT@exp@cs\ifx\MT@tempa @#1font@list}\relax\else
1042     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
1043       \MT@font \cename MT@tempa @#1font@list\endcsname
1044   \fi
1045 }
1046 \pdfTeX-def}\relax

```

\MT@lua@copy@font <#1> and <#2> are ‘select’ and ‘font’, respectively, <#3> is the font spec.

```

1047 \luaTeX-def\def\MT@lua@copyfont #1 #2 #3\@nil{%
1048 \luaTeX-def \global\expandafter\font\MT@font@copy=#3\relax}
1049 \pdfTeX-def\luaTeX-def

```

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 10 pt:

```

\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@fix@fontdimen@six
 \MT@dimen@six

If \fontdimen 6 is zero, character protrusion, spacing, kerning and tracking won’t work, and we could skip the settings (for example, the dsfont fonts don’t specify this dimension; this is probably a bug – the fourier and newpx/newtx packages have been fixed in the meantime). However, we can fix it ourselves – only tracking still doesn’t work (it seems that \letterspacefont uses the \fontdimen 6 from the original font). XeTeX doesn’t provide an equivalent to \pdffontsize, so we use the nominal size instead.

```

1050 \pdfTeX-def\luaTeX-def\xetex-def
1051 \def\MT@fix@fontdimen@six{%
1052   \ifnum\fontdimen6\MT@font=\z@
1053     \fontdimen6\MT@font=%
1054   \pdfTeX-def \pdffontsize\MT@font
1055   \luaTeX-def \MT@requires@luaTeX4{\pdffeedback fontsize}{\pdffontsize}\MT@font
1056   \xetex-def \MT@size pt
1057   \MT@info{Fixing zero \string\fontdimen 6 for font ‘\MT@font’\MessageBreak
1058     (new value: \the\fontdimen6\MT@font)}%
1059   \MT@gl@et@nc{\MT@font-fake6}\empty
1060   \fi
1061   \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
1062 }
1063 \pdfTeX-def\luaTeX-def\xetex-def

```

\MT@split@name Split up the font name (<#6> may be a protrusion/expansion context and/or a letterspacing amount). With fontspec we also need to remove its internal instance counter.

\MT@encoding

\MT@family

\MT@series 1064 *(*package)*

\MT@shape 1065 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%

\MT@size 1066 \def\MT@encoding{#1}%

1067 \ifMT@fontspec

1068 \edef\MT@family{\MT@scrubfeature#2()\relax}%

1069 \else

1070 \def\MT@family{#2}%

1071 \fi

1072 \def\MT@series {#3}%

1073 \def\MT@shape {#4}%

1074 \def\MT@size {#5}%

1075 \MT@fix@fontdimen@six

\MT@familyalias Alias family?

1076 \MT@ifdefined@n@TF{MT@MT@family @alias}%

1077 {\MT@let@cn\MT@familyalias{MT@MT@family @alias}}%

1078 {\let\MT@familyalias\@empty}%

1079 }

\MT@scrubfeature Remove one resp. all feature counters (fontspec).

\MT@scrubfeatures 1080 \def\MT@scrubfeature#1(#2)#3\relax{#1}

1081 \def\MT@scrubfeatures#1(#2)#3\relax{%

1082 #1%

1083 \ifx\relax#3\relax\else

1084 \MT@scrubfeatures#3\relax

1085 \fi

1086 }

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

\MT@feat

\MT@maybe@do 1087 \newif\ifMT@do

1088 \def\MT@maybe@do#1{%

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

1089 \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).

1090 \MT@dotrue

1091 \edef\@tempa{\csname MT@#1@setname\endcsname}%

1092 \MT@map@clist@n{font,encoding,family,series,shape,size}{%

1093 \MT@ifdefined@n@TF{MT@checklist@##1}%

1094 {\csname MT@checklist@##1\endcsname}%

1095 {\MT@checklist@{##1}}%

1096 {#1}%

1097 }%

1098 \else

1099 \MT@dofalse

1100 \fi

1101 \ifMT@do

\MT@feat stores the current feature.

1102 \def\MT@feat{#1}%

1103 \csname MT@set@#1@codes\endcsname

1104 \else

1105 \MT@ifstreq{#1}{tr}%

1106 {\let\MT@info@nottracking\MT@info@nottracking@}%

1107 {\MT@vinfo{... No \@nameuse{MT@abbr@#1}}}%

1108 \fi

```

1109 }

\MT@info@nottracking    To defer the message to after the font has actually been logged.
\MT@info@nottracking@ 1110 \let\MT@info@nottracking\relax
1111 \def\MT@info@nottracking@{\MT@vinfo{... No tracking}}

\MT@info@list
1112 <debug>\def\MT@info@list#1#2#3{\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: #2
1113 <debug> \ifx\#3\list empty\else \@nameuse{MT@#2}' #3 list\fi}}

\MT@checklist@    The generic test (<#1> is the axis, <#2> the feature, \@tempa contains the set name).
1114 \def\MT@checklist@#1#2{%
1115 <!debug> \MT@ifdefined@n@T
1116 <debug> \MT@ifdefined@n@TF
1117 {MT@#2list@#1@ \@tempa}{%

Begin a (neatly masqueraded) \expandafter orgy to test whether the font attribute
is in the list.
1118 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
1119 \csname MT@#1\expandafter\endcsname
1120 \csname MT@#2list@#1@ \@tempa\endcsname
1121 \ifMT@inlist@
1122 <debug>\MT@info@list{#2}{#1}{in}%
1123 \MT@dotrue
1124 \else
1125 <debug>\MT@info@list{#2}{#1}{not in}%
1126 \MT@dofalse
1127 \expandafter\MT@clist@break
1128 \fi
1129 }%

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.
1130 <debug> {\MT@info@list{#2}{#1}{}}%
1131 }

\MT@checklist@family    Also test for the alias font, if the original font is not in the list.
1132 \def\MT@checklist@family#1{%
1133 <!debug> \MT@ifdefined@n@T
1134 <debug> \MT@ifdefined@n@TF
1135 {MT@#1list@family@ \@tempa}{%
1136 \MT@exp@two@n\MT@in@clist
1137 \MT@family{\csname MT@#1list@family@ \@tempa\endcsname}%
1138 \ifMT@inlist@
1139 <debug>\MT@info@list{#1}{family}{in}%
1140 \MT@dotrue
1141 \else
1142 <debug>\MT@info@list{#1}{family}{not in}%
1143 \MT@dofalse
1144 \ifx\MT@familyalias\empty \else
1145 \MT@exp@two@n\MT@in@clist
1146 \MT@familyalias{\csname MT@#1list@family@ \@tempa\endcsname}%
1147 \ifMT@inlist@
1148 <debug> \MT@info@list{#1}{family alias}{in}%
1149 \MT@dotrue
1150 <debug>\else\MT@info@list{#1}{family alias}{not in}%
1151 \fi
1152 \fi
1153 \fi
1154 \ifMT@do \else
1155 \expandafter\MT@clist@break
1156 \fi
1157 }%
1158 <debug> {\MT@info@list{#1}{family}{}}%

```

```

1159 }
\MT@checklist@size    Test whether font size is in list of size ranges.
1160 \def\MT@checklist@size#1{%
1161   \!debug \MT@ifdefined@n@T
1162   \!debug \MT@ifdefined@n@TF
1163   {MT@#1list@size@%tempa}%
1164   \MT@exp@cs\MT@in@rlist{MT@#1list@size@%tempa}%
1165   \ifMT@inlist@
1166   \!debug \MT@info@list{#1}{size}{in}%
1167   \MT@dotrue
1168   \else
1169   \!debug \MT@info@list{#1}{size}{not in}%
1170   \MT@dofalse
1171   \expandafter\MT@clist@break
1172   \fi
1173   }%
1174   \!debug { \MT@info@list{#1}{size}{}}%
1175 }

```

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

```

1176 \def\MT@checklist@font#1{%
1177   \!debug \MT@ifdefined@n@T
1178   \!debug \MT@ifdefined@n@TF
1179   {MT@#1list@font@%tempa}%

```

Since \MT@font may be appended with context and/or letterspacing specs, we construct the name from the font characteristics.

```

1180   \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
1181   \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
1182   \@tempb \csname MT@#1list@font@%tempa\endcsname
1183   \ifMT@inlist@
1184   \!debug \MT@info@list{#1}{font}{in}%
1185   \expandafter\MT@clist@break
1186   \else
1187   \!debug \MT@info@list{#1}{font}{not in}%
1188   \MT@dofalse
1189   \fi
1190   }%
1191   \!debug { \MT@info@list{#1}{font}{}}%
1192 }

```

14.2.1 Protrusion

\ifMT@nofamily Info for settings that are not family-specific. (Warnings seem to be too irritating.)
The switch is set in \MT@next@listname.

```

1193 \newif\ifMT@nofamily
1194 \!package

```

\MT@protrusion Set up for protrusion?

```

1195 \!pdfTeX-def|xetex-def|luatex-def
1196 \def\MT@protrusion{\MT@maybe@do{pr}}

```

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

```

1197 \def\MT@set@pr@codes{%
1198   \MT@nofamilyfalse

```

Check whether and if, which list should be applied to the current font. If family-specific settings don't exist, we write it to the log (for each encoding).

```

1199   \MT@if@list@exists{%
1200     \ifMT@nofamily

```

```

1201 \MT@ifdefined@n@TF{\MT@encoding-\MT@family-settings}\relax{%
1202 \MT@info@n{Loading generic protrusion settings for font family\MessageBreak
1203 \MT@family' (encoding: \MT@encoding).\MessageBreak
1204 For optimal results, create family-specific settings.\MessageBreak
1205 See the microtype manual for details}%
1206 \MT@glet@nc{\MT@encoding-\MT@family-settings}\@empty
1207 }%
1208 \fi
1209 \MT@get@opt
1210 \MT@reset@pr@codes

```

Get the name of the inheritance list and parse it.

```

1211 \MT@get@inh@list
Set an input encoding?
1212 \MT@set@inputenc{c}%

```

Load additional lists?

```

1213 \MT@load@list\MT@pr@c@name
1214 \MT@set@listname

```

Load the main list.

```

1215 \MT@let@cn\@tempc{\MT@pr@c@\MT@pr@c@name}%
1216 \expandafter\MT@set@codes\@tempc,\relax,%
1217 }\MT@reset@pr@codes
1218 }

```

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

```

1219 \def\MT@set@all@pr#1#2{%
1220 <debug>\MT@info@n{3}{-- lp/rp: setting all to #1/#2}%
1221 \let\MT@temp\@empty
1222 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lpcode\MT@font\@tempcnta=#1}}%
1223 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rpcode\MT@font\@tempcnta=#2}}%
1224 \MT@do@font\MT@temp
1225 }

```

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

```

1226 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@z@}
1227 \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 margin kerns. This will be activated in \MT@set@tr@codes.

```

1228 \def\MT@the@pr@code{\@tempcntb}
1229 <pdfTeX-def|luatex-def>
1230 <pdfTeX-def>\MT@requires@pdfTeX6
1231 <luatex-def>\MT@requires@luatex3
1232 {\def\MT@the@pr@code@tr{%
1233 \numexpr\@tempcntb+\MT@letterspace@/2\relax
1234 }
1235 }\relax
1236 </pdfTeX-def|luatex-def>

```

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

```

1237 \def\MT@set@codes#1,{%
1238 \ifx\relax#1\@empty\else
1239 \MT@split@codes #1==\relax
1240 \expandafter\MT@set@codes
1241 \fi
1242 }

```

\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.

```

1243 \def\MT@split@codes#1=#2=#3\relax{%
1244   \def\@tempa{#1}%
1245   \ifx\@tempa\@empty \else
1246     \MT@get@slot
1247     (pdfTeX-def|LaTeX-def) \ifnum\MT@char > \m@ne
1248     (XeTeX-def) \ifx\MT@char\@empty \else
1249       \MT@get@char@unit
1250       \csname MT@MT@feat @split@val\endcsname#2\relax
1251     \fi
1252   \fi
1253 }

```

\MT@pr@split@val

```

1254 \def\MT@pr@split@val#1,#2\relax{%
1255   \def\@tempb{#1}%
1256   \MT@ifempty\@tempb\relax{%
1257     \MT@scale@to@em
1258     \lcode\MT@font\MT@char=\MT@the@pr@code
1259     (debug)\MT@info@n1{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char\space: [#1]}%
1260   }%
1261   \def\@tempb{#2}%
1262   \MT@ifempty\@tempb\relax{%
1263     \MT@scale@to@em
1264     \rprcode\MT@font\MT@char=\MT@the@pr@code
1265     (debug)\MT@info@n1{4}{;;; rp (\MT@char): \number\rprcode\MT@font\MT@char\space: [#2]}%
1266   }%

```

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

```

1267 \MT@ifdefined@c@T\MT@pr@inh@name{%
1268   \MT@ifdefined@n@T{\MT@inh@\MT@pr@inh@name @\MT@char @}{%
1269     \MT@exp@cs\MT@map@tlist@c
1270     {\MT@inh@\MT@pr@inh@name @\MT@char @}%
1271     \MT@set@pr@heirs
1272   }%
1273 }%
1274 }

```

\MT@scale@to@em

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

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

```

1275 (pdfTeX-def)\MT@requires@pdfTeX3{
1276 \def\MT@scale@to@em{%
1277   \@tempcntb=\MT@count\relax

```

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

```

1278   \MT@scale\@tempcntb \@tempb \MT@dimen@six
1279   \ifnum\@tempcntb=\z@ \else
1280     \MT@scale@factor
1281   \fi
1282 }

```


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

```
1283 \def\MT@get@charwd{%
1284   \ifnum\MT@char<\z@
1285     \MT@count=\fontcharwd\MT@font\MT@char\relax
1286   \else
1287     \setbox\z@=\hbox{\MT@font \char\MT@char}%
1288     \MT@count=\wd\z@
1289   \fi
1290 }
1291 \ifnum\MT@char<\z@
1292   \setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char}%
1293   \MT@count=\wd\z@
1294 \else
1295   \MT@count=\fontcharwd\MT@font\MT@char\relax
1296 \fi
1297 }
1298 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1299 }
```

`\MT@char` contains a slot number (legacy fonts), a Unicode number, or a glyph name (if `\MT@char@` is negative).

```
1290 \ifnum\MT@char<\z@
1291   \setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char}%
1292   \MT@count=\wd\z@
1293 \else
1294   \MT@count=\fontcharwd\MT@font\MT@char\relax
1295 \fi
1296 }
1297 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1298 }
1299 }
```

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`.

```
1300 \ifnum\MT@char<\z@
1301   \MT@requires@pdftex6{
1302     \g@addto@macro\MT@get@charwd{%
1303       \MT@ifdefined@cT\MT@letterspace@
1304       {\advance\MT@count-\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1305     }
1306   }\relax
1307 }
```

No adjustment with versions 0.14f and 0.14g.

```
1308 \def\MT@scale@to@em{%
1309   \MT@count=\@tempb\relax
1310   \ifnum\MT@count=\z@ \else
1311     \MT@scale@factor
1312   \fi
1313 }
```

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

```
1314 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1315 }
1316 \ifnum\MT@char<\z@
1317   \setbox\z@=\hbox{\MT@font \char\MT@char}%
1318   \MT@count=\wd\z@
1319 \else
1320   \MT@count=\fontcharwd\MT@font\MT@char\relax
1321 \fi
1322 }
```

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

```
1318 \ifnum\MT@fontdimen<\z@
1319   \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1320     \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1321     You should use a different 'unit' for \MT@curr@list@name}%
1322   \else
1323     \MT@count=\fontdimen#1\MT@font
1324   \fi
1325 }
1326 \ifnum\MT@fontdimen<\z@
1327   \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1328     \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1329     You should use a different 'unit' for \MT@curr@list@name}%
1330   \else
1331     \MT@count=\fontdimen#1\MT@font
1332   \fi
1333 }
```

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

```
1328 \def\MT@info@missing@char{%
```

```

1329 \MT@info@n1{Character `the\MT@toks'
1330 ^^X \ifnum\MT@char@<\z@ is missing\else
1331 ^^X \iffontchar\MT@font\MT@char@
1332 has a width of 0pt
1333 ^^X \else is missing\fi\fi
1334 ^^Q \MessageBreak (it's probably missing)
1335 \MessageBreak in font `the\MT@font'.\MessageBreak
1336 Ignoring protrusion settings for this character}%
1337 }

```

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

```

1338 \def\MT@scale@factor{%
1339 \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1340 \expandafter\MT@scale\expandafter \@tempcntb
1341 \csname MT@\MT@feat @factor@\endcsname \@m
1342 \fi
1343 \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
1344 \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1345 \else
1346 \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1347 \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1348 \fi
1349 \fi
1350 }

```

\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.

```

1351 \def\MT@warn@code@too@large#1{%
1352 \@tempcnta=#1\relax
1353 \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1354 \expandafter\MT@scale\expandafter \@tempcnta\expandafter
1355 \@m \csname MT@\MT@feat @factor@\endcsname
1356 \fi
1357 \MT@scale\@tempcnta \MT@dimen@six \MT@count
1358 \MT@warning@n1{The \nameuse{MT@abbr@\MT@feat} code \@tempb\space
1359 is too large for character\MessageBreak
1360 `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1361 Setting it to the maximum of \number\@tempcnta}%
1362 \@tempcntb=#1\relax
1363 }

```

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

```

1364 \def\MT@get@opt{%
1365 \MT@set@listname

```

\MT@pr@factor@ Apply a factor?

```

\MT@sp@factor@ 1366 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 1367 \MT@let@nn{MT@\MT@feat @factor@}
1368 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1369 \MT@vinfo{... : Multiplying \nameuse{MT@abbr@\MT@feat} codes by
1370 \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1371 }{%
1372 \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1373 }%

```

\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@ 1374 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1375 \MT@let@nn{MT@\MT@feat @unit@}%
1376 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%

```

```

1377 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
1378 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1379         relative to character widths}%
1380 \else
1381 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
1382 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1383         relative to width of space}%
1384 \fi
1385 \fi
1386 }{%
1387 \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit@}%
1388 }%

```

\MT@get@space@unit The codes are either relative to character widths, or to a fixed width. For spacing
 \MT@get@char@unit 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.

```

1389 \let\MT@get@char@unit\relax
1390 \let\MT@get@space@unit\@gobble
1391 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
1392 \let\MT@get@char@unit\MT@get@charwd
1393 \else
1394 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
1395 \let\MT@get@space@unit\MT@get@font@dimen
1396 \else
1397 \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}%
1398 \fi
1399 \fi

```

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

```

1400 \MT@ifdefined@nT{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}}{%
1401 \csname MT@preset@\MT@feat\endcsname
1402 \MT@let@nc{MT@reset@\MT@feat @codes}\relax
1403 }%
1404 }

```

\MT@get@unit If unit contains an em or ex, we use the corresponding \fontdimen to obtain the
 \MT@get@unit@ 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.

```

1405 \def\MT@get@unit#1{%
1406 \expandafter\MT@get@unit@#1 e!\@nil
1407 \ifx\x\@empty\else\let#1\x\fi
1408 \@defaultunits\@tempdima#1 pt\relax\@nnil
1409 \ifdim\@tempdima=\z@
1410 \MT@warning@n1{%
1411 Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
1412 width. Setting factors of list \@nameuse{MT@\MT@feat @c@name}'\MessageBreak
1413 relative to character widths instead}%
1414 \let#1\@empty
1415 \let\MT@get@char@unit\MT@get@charwd
1416 \else
1417 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
1418 to \the\@tempdima}%
1419 \MT@count=\@tempdima\relax
1420 \fi
1421 }
1422 \def\MT@get@unit@#1e#2#3\@nil{%
1423 \ifx\#3\\\let\x\@empty \else
1424 \if m#2%
1425 \edef\x{#1\fontdimen6\MT@font}%
1426 \else
1427 \if x#2%
1428 \edef\x{#1\fontdimen5\MT@font}%
1429 \fi

```

```

1430   \fi
1431   \fi
1432 }

\MT@set@inputenc    The configurations may be under the regime of an input encoding.
1433 \def\MT@set@inputenc#1{%
\MT@cat    We remember the current category (c or inh), in case of warnings later.
1434   \def\MT@cat{#1}%
1435   \edef\@tempa{\MT@MT@feat @#1\csname MT@\MT@feat @#1\name\endcsname @inputenc}%
1436   \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1437 }

\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.
1438 \MT@addto@setup{%
1439   \ifpackageloaded{inputenc}{%
1440     \ifpackageafter{inputenc}{2006/02/22}{%
1441       \def\MT@set@inputenc@{%
1442         \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1443         \MT@load@inputenc
1444       }%
1445     }%
1446     \let\MT@set@inputenc@\MT@load@inputenc
1447   }%
1448 }%
1449   \def\MT@set@inputenc@{%
1450     \MT@warning@n{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1451       \MessageBreak package isn't loaded. Ignoring input encoding}%
1452   }%
1453 }%
1454 }

\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.
1455 \def\MT@load@inputenc{%
1456   \MT@cfg@catcodes
1457   <debug>\MT@info@n{1}{loading input encoding: \@nameuse{\@tempa}}%
1458   \inputencoding{\@nameuse{\@tempa}}%
1459 }
1460 </package>

\MT@set@pr@heirs    Set the inheriting characters.
1461 <pdfTeX-def>|xetex-def|<luatex-def>
1462 \def\MT@set@pr@heirs#1{%
1463   \lcode\MT@font #1=\lcode\MT@font\MT@char\relax
1464   \rcode\MT@font #1=\rcode\MT@font\MT@char\relax
1465   <debug>\MT@info@n{2}{-- heir of \MT@char: #1}%
1466   <debug>\MT@info@n{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char\space/%
1467   <debug>                                     \number\rcode\MT@font\MT@char\space}%
1468 }

\MT@preset@pr    Preset characters. Presetting them relative to their widths is not allowed.
\MT@preset@pr@
1469 \def\MT@preset@pr{%
1470   \expandafter\expandafter\expandafter\MT@preset@pr@
1471   \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1472 }
1473 \def\MT@preset@pr@#1,#2\@nil{%
1474   \ifx\MT@pr@unit@\@empty
1475     \MT@warn@preset@t@width{pr}%
1476     \let\MT@preset@aux\MT@preset@aux@factor
1477   \else
1478     \def\MT@preset@aux{\MT@preset@aux@space2}%
1479   \fi

```

```

1480 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1481 \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1482 \MT@set@all@pr\@tempa\@tempb
1483 }

```

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

```

\MT@preset@aux@factor 1484 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 1485 \@tempcntb=#1\relax
1486 \MT@scale@factor
1487 \edef#2{\number\@tempcntb}%
1488 }
1489 \def\MT@preset@aux@space#1#2#3{%
1490 \def\@tempb{#2}%
1491 \MT@get@space@unit#1%
1492 \MT@scale@to@em
1493 \edef#3{\number\@tempcntb}%
1494 }

```

\MT@warn@preset@tewidth

```

1495 \def\MT@warn@preset@tewidth#1{%
1496 \MT@warning@nl{%
1497 Cannot preset characters relative to their widths\MessageBreak
1498 for \@nameuse{MT@abbr#1} list ~\@nameuse{MT@#1@c@name}'. Presetting them%
1499 \MessageBreak relative to lem instead}%
1500 }
1501 </pdfTeX-def|xetex-def|luatex-def>

```

14.2.2 Expansion

\MT@expansion Set up for expansion?

```

1502 <*pdfTeX-def|luatex-def>
1503 \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).

```

1504 \def\MT@set@ex@codes@{%
1505 \MT@if@list@exists{%
1506 \MT@get@ex@opt
1507 \let\MT@get@char@unit\relax
1508 \MT@reset@ef@codes
1509 \MT@get@inh@list
1510 \MT@set@inputenc{c}%
1511 \MT@load@list\MT@ex@c@name
1512 \MT@set@listname
1513 \MT@let@cn\@tempc{MT@ex@c@\MT@ex@c@name}%
1514 \expandafter\MT@set@codes\@tempc,\relax,%
1515 \MT@expandfont
1516 }\relax
1517 }
1518 </pdfTeX-def|luatex-def>

```

\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.

```

1519 <package>\newif\ifMT@nonselected
1520 <*pdfTeX-def|luatex-def>
1521 \def\MT@set@ex@codes@n{%
1522 \MT@nonselectedtrue

```

```

1523 \MT@if@list@exists
1524 \MT@get@ex@opt
1525 {%
1526 \let\MT@stretch@ \MT@stretch
1527 \let\MT@shrink@ \MT@shrink
1528 \let\MT@step@ \MT@step
1529 \let\MT@auto@ \MT@auto
1530 \let\MT@ex@factor@ \MT@ex@factor
1531}%
1532 \MT@reset@ef@codes
1533 \MT@expandfont
1534 \MT@nonselectedfalse
1535 }

```

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

```
1536 \let\MT@set@ex@codes\MT@set@ex@codes@n
```

`\MT@expandfont` Expand the font. For some reason, older LuaTeX versions freeze if the autoexpand modifier is missing. Can't be bothered to find out why. For newer versions, we could also use the function `font.setexpansion`, or, in the future, `luaotfload`'s expansion font feature.

```

1537 <luatex-def>
1538 \MT@requires@luatex3{
1539 \MT@requires@luatex4{\let\pdffontexpand\expandglyphsinfont}\relax
1540 \ifnum\luatexversion<79
1541 \def\MT@expandfont{%
1542 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ autoexpand\relax
1543 }
1544 \else
1545 \def\MT@expandfont{%
1546 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@\relax
1547 }
1548 \fi
1549 }{
1550 </luatex-def>
1551 \def\MT@expandfont{%
1552 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1553 }
1554 <luatex-def>}

```

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

```

1555 \def\MT@set@all@ex#1{%
1556 <debug>\MT@info@n1{3}{-- ex: setting all to \number#1}%
1557 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1558 }
1559 \def\MT@reset@ef@codes@{\MT@set@all@ex\MT@ex@factor@}

```

`\MT@reset@ef@codes` However, this is only necessary for pdfTeX versions prior to 1.20, or LuaTeX < 0.90 (actually, I think, 0.87).

```

1560 <pdfTeX-def>\MT@requires@pdfTeX4
1561 <luatex-def>\MT@requires@luatex5
1562 {
1563 \def\MT@reset@ef@codes{%
1564 \ifnum\MT@ex@factor@=\@m \else
1565 \MT@reset@ef@codes@
1566 \fi
1567 }
1568 }{
1569 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1570 }

```

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

```

1571 \def\MT@ex@split@val#1\relax{%
1572   \@tempcntb=#1\relax

```

Take an optional factor into account.

```

1573   \ifnum\MT@ex@factor@=\@m \else
1574     \MT@scale\@tempcntb \MT@ex@factor@ \@m
1575   \fi
1576   \ifnum\@tempcntb > \MT@ex@max
1577     \MT@warn@ex@too@large\MT@ex@max
1578   \else
1579     \ifnum\@tempcntb < \MT@ex@min
1580       \MT@warn@ex@too@large\MT@ex@min
1581     \fi
1582   \fi
1583   \efcode\MT@font\MT@char=\@tempcntb
1584   <debug>\MT@info@n1{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%

```

Heirs, heirs, I love thy heirs.

```

1585   \MT@ifdefined@c@T\MT@ex@inh@name{%
1586     \MT@ifdefined@n@T{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1587       \MT@exp@cs\MT@map@tlist@c{\MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
1588     }%
1589   }%
1590 }

```

\MT@warn@ex@too@large

```

1591 \def\MT@warn@ex@too@large#1{%
1592   \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1593     character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1594     Setting it to the maximum of \number#1}%
1595   \@tempcntb=#1\relax
1596 }

```

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

```

\MT@ex@factor@ 1597 \def\MT@get@ex@opt{%
\MT@stretch@   \MT@set@listname
1598   \MT@stretch@ 1599   \MT@ifdefined@n@TF{\MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@    \MT@let@cn\MT@ex@factor@{\MT@ex@c@\MT@ex@c@name @factor}%
1600   \MT@shrink@ 1601   \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@step@      1602   }%
1603   \MT@step@    1603   \let\MT@ex@factor@\MT@ex@factor
1604   \MT@step@    1604   }%
1605   \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1606   \MT@get@ex@opt@{shrink}{Setting shrink limit to \number\MT@shrink@}%
1607   \MT@get@ex@opt@{step}{Setting expansion step to \number\MT@step@}%
1608   <luatex-def> \MT@requires@luatex3\relax%
1609   \MT@get@ex@opt@{auto}{\MT@ifstreq{\MT@auto@}{autoexpand}{En}{Dis}abling automatic expansion}%
1610   <luatex-def> }%
1611   \MT@ifdefined@n@T{\MT@ex@c@\MT@ex@c@name @preset}{%
1612     \MT@preset@ex
1613     \let\MT@reset@ef@codes\relax
1614   }%
1615 }

```

\MT@get@ex@opt@

```

1616 \def\MT@get@ex@opt@#1#2{%
1617   \MT@ifdefined@n@TF{\MT@ex@c@\MT@ex@c@name @#1}{%
1618     \MT@let@nn{\MT@#1@}{\MT@ex@c@\MT@ex@c@name @#1}%
1619     \MT@vinfo{... : #2}%
1620   }%
1621   \MT@let@nn{\MT@#1@}{\MT@#1}%
1622 }%
1623 }

```

\MT@set@ex@heirs

```

1624 \def\MT@set@ex@heirs#1{%
1625   \efcode\MT@font#1=\efcode\MT@font\MT@char
1626   (debug)\MT@info{n}{2}{-- heir of \MT@char: #1}%
1627   (debug)\MT@info{n}{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1628 }

```

\MT@preset@ex

```

1629 \def\MT@preset@ex{%
1630   \@tempcntb=\csname MT@ex@cc\MT@ex@ccname @preset\endcsname\relax
1631   \MT@scale@factor
1632   \MT@set@all@ex\@tempcntb
1633 }
1634 (pdfTeX-def|luTeX-def)

```

14.2.3 Interword spacing (glue)

\MT@spacing Adjustment of interword spacing? Only works with pdfTeX.

```

1635 (*pdfTeX-def)
1636 \MT@requires@pdfTeX6{
1637 \def\MT@spacing{\MT@maybe@do{sp}}

```

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

```

1638 \def\MT@set@sp@codes{%
1639   \MT@if@list@exists{%
1640     \MT@get@opt
1641     \MT@reset@sp@codes
1642     \MT@get@inh@list
1643     \MT@set@inputenc{c}%
1644     \MT@load@list\MT@sp@ccname
1645     \MT@set@listname
1646     \MT@let@cn\@tempc{\MT@sp@cc\MT@sp@ccname}%
1647     \expandafter\MT@set@codes\@tempc,\relax,%
1648   }\MT@reset@sp@codes
1649 }

```

\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).

```

1650 \def\MT@sp@split@val#1,#2,#3\relax{%
1651   \def\@tempb{#1}%
1652   \MT@ifempty\@tempb\relax{%
1653     \MT@get@space@unit2%
1654     \MT@scale@to@em
1655     \knbscode\MT@font\MT@char=\@tempcntb
1656     (debug)\MT@info{n}{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1657   }%
1658   \def\@tempb{#2}%
1659   \MT@ifempty\@tempb\relax{%
1660     \MT@get@space@unit3%
1661     \MT@scale@to@em
1662     \stbscode\MT@font\MT@char=\@tempcntb
1663     (debug)\MT@info{n}{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1664   }%
1665   \def\@tempb{#3}%
1666   \MT@ifempty\@tempb\relax{%
1667     \MT@get@space@unit4%
1668     \MT@scale@to@em
1669     \shbscode\MT@font\MT@char=\@tempcntb
1670     (debug)\MT@info{n}{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1671   }%
1672   \MT@ifdefined@c@T\MT@sp@inh@name{%
1673     \MT@ifdefined@nT{\MT@inh@MT@sp@inh@name @\MT@char @}{%
1674       \MT@exp@cs\MT@map@tlist@c{\MT@inh@MT@sp@inh@name @\MT@char @}\MT@set@sp@heirs
1675     }%

```



```

1676 }%
1677 }

\MT@set@sp@heirs
1678 \def\MT@set@sp@heirs#1{%
1679 \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1680 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1681 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1682 (debug)\MT@info{n1}{2}{-- heir of \MT@char: #1}%
1683 (debug)\MT@info{n1}{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1684 (debug) \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1685 }

\MT@set@all@sp
\MT@reset@sp@codes 1686 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1687 (debug)\MT@info{n1}{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1688 \let\MT@temp@empty
1689 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
1690 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1691 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1692 \MT@do@font\MT@temp
1693 }
1694 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1695 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1696 \def\MT@preset@sp{%
1697 \expandafter\expandafter\expandafter\MT@preset@sp@
1698 \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1699 }
1700 \def\MT@preset@sp@#1,#2,#3\@nil{%
1701 \ifx\MT@sp@unit@\empty
1702 \MT@warn@preset@twidth{sp}%
1703 \MT@ifempty{#1}{\let\@tempa\empty}{\MT@preset@aux@factor{#1}\@tempa}%
1704 \MT@ifempty{#2}{\let\@tempc\empty}{\MT@preset@aux@factor{#2}\@tempc}%
1705 \MT@ifempty{#3}{\let\@tempb\empty}{\MT@preset@aux@factor{#3}\@tempb}%
1706 \else
1707 \MT@ifempty{#1}{\let\@tempa\empty}{\MT@preset@aux@space2{#1}\@tempa}%
1708 \MT@ifempty{#2}{\let\@tempc\empty}{\MT@preset@aux@space3{#2}\@tempc}%
1709 \MT@ifempty{#3}{\let\@tempb\empty}{\MT@preset@aux@space4{#3}\@tempb}%
1710 \fi
1711 \MT@set@all@sp\@tempa\@tempc\@tempb
1712 }
1713 }\relax

```

14.2.4 Additional kerning

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

```

1714 \MT@requirespdfTeX{
1715 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

1716 \def\MT@set@kn@codes{%
1717 \MT@if@list@exists{%
1718 \MT@get@opt
1719 \MT@reset@kn@codes
1720 \MT@get@inh@list
1721 \MT@set@inputenc{c}%
1722 \MT@load@list\MT@kn@c@name
1723 \MT@set@listname
1724 \MT@let@cn\@tempc{\MT@kn@c@\MT@kn@c@name}%
1725 \expandafter\MT@set@codes\@tempc,\relax,%
1726 }\MT@reset@kn@codes

```

1727 }

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

```

1728 \def\MT@kn@split@val#1,#2\relax{%
1729   \def\@tempb{#1}%
1730   \MT@ifempty\@tempb\relax{%
1731     \MT@get@space@unit2%
1732     \MT@scale@to@em
1733     \knbcode\MT@font\MT@char=\@tempcntb
1734   <debug>\MT@info@n1{4}{;;; knbc (\MT@char): \number\knbcode\MT@font\MT@char: [#1]}%
1735   }%
1736   \def\@tempb{#2}%
1737   \MT@ifempty\@tempb\relax{%
1738     \MT@get@space@unit2%
1739     \MT@scale@to@em
1740     \knacode\MT@font\MT@char=\@tempcntb
1741   <debug>\MT@info@n1{4}{;;; knac (\MT@char): \number\knacode\MT@font\MT@char: [#2]}%
1742   }%
1743   \MT@ifdefined@c@T\MT@kn@inh@name{%
1744     \MT@ifdefined@n@T{\MT@inh@\MT@kn@inh@name @\MT@char @}{%
1745       \MT@exp@cs\MT@map@tlist@c{\MT@inh@\MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1746     }%
1747   }%
1748 }
```

\MT@set@kn@heirs

```

1749 \def\MT@set@kn@heirs#1{%
1750   \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1751   \knacode\MT@font#1=\knacode\MT@font\MT@char
1752   <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1753   <debug>\MT@info@n1{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1754   <debug>                                     \number\knacode\MT@font\MT@char}%
1755 }
```

\MT@set@all@kn

```

\MT@reset@kn@codes 1756 \def\MT@set@all@kn#1#2{%
1757   <debug>\MT@info@n1{3}{-- knac/knbc: setting all to #1/#2}%
\MT@reset@kn@codes@ 1758   \let\MT@temp\@empty
1759   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbcode\MT@font\@tempcnta=#1\relax}}%
1760   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knacode\MT@font\@tempcnta=#2\relax}}%
1761   \MT@do@font\MT@temp
1762 }
1763 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1764 \let\MT@reset@kn@codes\relax
```

\MT@preset@kn

```

\MT@preset@kn@ 1765 \def\MT@preset@kn{%
1766   \expandafter\expandafter\expandafter\MT@preset@kn@
1767   \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1768 }
1769 \def\MT@preset@kn@#1,#2\@nil{%
1770   \ifx\MT@kn@unit@\@empty
1771     \MT@warn@preset@twidth{kn}%
1772     \let\MT@preset@aux\MT@preset@aux@factor
1773   \else
1774     \def\MT@preset@aux{\MT@preset@aux@space2}%
1775   \fi
1776   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1777   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1778   \MT@set@all@kn\@tempa\@tempb
1779 }
1780 }\relax
1781 </pdfTeX-def>
```

14.2.5 Tracking

This only works with pdfTeX 1.40 or LuaTeX 0.62.

```
1782 <pdfTeX-def|luatex-def>
1783 <pdfTeX-def>\MT@requires@pdfTeX6
1784 <luatex-def>\MT@requires@luatex3
1785 {
```

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

```
\MT@tr@font@list 1786 \let\MT@tr@font@list\empty
1787 \def\MT@tracking@{%
1788   \MT@exp@one@n\MT@in@cl@list\MT@font\MT@tr@font@list
1789   \ifMT@inlist@else
1790     \MT@maybe@do{tr}%
1791     \ifMT@do@else
1792       \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1793     \fi
1794   \fi
1795 }
1796 </pdfTeX-def|luatex-def>
1797 <pdfTeX-def|luatex-def|letterspace>\let\MT@tracking
1798 <pdfTeX-def|luatex-def> \MT@tracking@
1799 <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.

Tracking won't work if the original font's \fontdimen 6 is zero, in which case we issue a warning (once for every font).

```
1800 <pdfTeX-def|luatex-def|letterspace>
1801 \def\MT@set@tr@codes{%
1802 <pdfTeX-def|luatex-def>
1803   \MT@vinfo{Tracking font '\MT@font'\on@line}%
1804   \MT@i@f@defined@n@TF{\MT@font-fake6}{%
1805     \expandafter\ifx\csname\MT@font-fake6\endcsname\empty
1806       \MT@warning@n1{%
1807         Font '\MT@font' does not specify its\MessageBreak
1808         \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
1809         \@nameuse{MT@abbr@MT@feat} will not work with this font}%
1810       \MT@glet@nc{\MT@font-fake6}\relax
1811     \fi
1812   }{%
1813     \MT@i@f@list@exists
1814     \MT@get@tr@opt
1815     \relax
1816 </pdfTeX-def|luatex-def>
1817   \MT@i@f@defined@c@TF\MT@letterspace@relax{\let\MT@letterspace@MT@letterspace}%
1818   \ifnum\MT@letterspace@=\z@
```

Zero tracking requires special treatment.

```
1819   \MT@set@tr@zero
1820   \else
1821 <pdfTeX-def|luatex-def> \MT@vinfo{... Tracking by \number\MT@letterspace@}%
```

Letterspacing only works in PDF mode.

```
1822   \MT@warn@tracking@DVI
```

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

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.

```

1823 \xdef\MT@lsfont{\csname\expandafter\string\font@name
1824 \number\MT@letterspace@ls\endcsname}%
1825 \expandafter\ifx\MT@lsfont\relax
1826 <debug>\MT@info{n1}{1}{... new letterspacing instance}%

```

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

```

1827 \MT@get@ls@basefont

```

luaotfload provides the faux font feature kernfactor, which we will use when dealing with non-legacy fonts, as it is less problematic and faster than the pdfTeX primitive \letterspacefont.

```

1828 <*luatex-def|letterspace>
1829 \MT@if@luaotf@font{%
1830 <luatex-def&debug>\MT@info{n1}{1}{... luaotf font: \MessageBreak
1831 <luatex-def&debug> \expandafter\fontname\font@name}%
1832 \ifnum\MT@letterspace@<z@\def\MT@minus{-}\else\let\MT@minus\empty\fi
1833 \global\expandafter\font\MT@lsfont=%
1834 \expandafter\MT@exp@two@c\expandafter\MT@ls@fontspec@font
1835 \expandafter\fontname\expandafter\font@name\space \@nil
1836 }{%
1837 </luatex-def|letterspace>
1838 <luatex-def&debug>\MT@info{n1}{1}{... legacy font}%
1839 \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@
1840 <luatex-def|letterspace> }%

```

Scale interword spacing (not configurable in letterspace).

```

1841 <*pdfTeX-def|luatex-def>
1842 \MT@ifdefined@c@TF\MT@tr@ispace
1843 {\let@tempa\MT@tr@ispace}%
1844 {\edef\@tempa{\MT@letterspace@*,,}}%
1845 \MT@ifdefined@c@TF\MT@tr@ospace
1846 {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1847 {\edef\@tempa{\@tempa,,,}}%
1848 \expandafter\MT@tr@set@space\@tempa,%
1849 </pdfTeX-def|luatex-def>
1850 <*letterspace>
1851 % spacing = <letterspace amount>*,,
1852 \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1853 * \fontdimen2\MT@lsfont/1000\relax
1854 </letterspace>

```

Adjust outer kerning (microtype only).

```

1855 <*pdfTeX-def|luatex-def>
1856 \MT@ifdefined@c@TF\MT@tr@okern{\let@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
1857 \expandafter\MT@tr@set@okern\@tempa,%

```

Disable ligatures (not configurable in letterspace).

```

1858 \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1859 </pdfTeX-def|luatex-def>
1860 <*letterspace>
1861 % no ligatures = {f}
1862 \tagcode\MT@lsfont`f=\m@ne
1863 </letterspace>

```

Adjust protrusion values now, and maybe later (in \MT@pr@split@val) (not for LuaTeX, though, where letterspacing does not interfere with protrusion).

```

1864 <luatex-def|letterspace> \MT@if@luaotf@font\relax{%
1865 <debug>\MT@info{n1}{2}{... compensating for tracking (\number\MT@letterspace@)}%
1866 \MT@do@font{\lpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
1867 \rprcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
1868 \let\MT@the@pr@code\MT@the@pr@code@tr
1869 <luatex-def|letterspace> }%
1870 \fi

```

Finally, let the letterspaced font propagate. With LuaTeX, we also need to load.

```
1871 \aftergroup\MT@set@lsfont
1872 <pdfTeX-def|luaTeX-def> \let\MT@font\MT@lsfont
1873 <luaTeX-def> \MT@if@luaotf@font\MT@font\relax
```

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

```
\MT@curr@ls 1874 \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
1875 \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.

```
1876 <*pdfTeX-def|luaTeX-def>
1877 \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
1878 \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1879 \MT@tr@outer@l
1880 </pdfTeX-def|luaTeX-def>
```

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

```
1881 \ifx\MT@ls@adjust@empty
1882 <letterspace> % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}
1883 \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1884 \MT@ls@outer@k
```

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

```
1885 <*pdfTeX-def|luaTeX-def>
1886 \else
1887 \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
1888 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1889 \ifdim\MT@outer@kern=\z@ \else \MT@ls@outer@k \fi
1890 \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
1891 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1892 </pdfTeX-def|luaTeX-def>
1893 <*letterspace>
1894 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1895 \MT@afteraftergroup{%
1896 \MT@set@curr@ok
1897 \noexpand\MT@ls@outer@k
1898 }%
1899 </letterspace>
1900 \fi
1901 <*pdfTeX-def|luaTeX-def>
```

`\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).

```
1902 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
```

Stuff to be done after the letterspace group. The `letterspace` package only adjusts the kerning.

```
1903 \MT@afteraftergroup{%
1904 \MT@set@curr@os
1905 \MT@set@curr@ok
1906 \noexpand\MT@tr@outer@r
1907 }%
1908 </pdfTeX-def|luaTeX-def>
1909 \fi
1910 <pdfTeX-def|luaTeX-def> }%
1911 }
```

`\MT@afteraftergroup` This helper macro carries stuff outside of the current group to the end of the next group, but will then respect grouping, which is crucial for nested letterspacing.

(Following an idea of Will Robertson.)

```

1912 \def\MT@afteraftergroup#1{%
1913 <!\letterspace> \MT@maybe@gobble@with@tikz{%
1914   \MT@ifdefined@n@TF{MT@aftergroup@number\currentgrouplevel}\relax{%
1915     \MT@exp@cs\xdef{MT@aftergroup@number\currentgrouplevel}%
1916     { \MT@exp@cs\MT@glet{MT@aftergroup@number\currentgrouplevel}\noexpand\undefined#1}%
1917     \expandafter\aftergroup\expandafter\aftergroup\MT@exp@cs\aftergroup
1918     {MT@aftergroup@number\currentgrouplevel}%
1919   }%
1920 <!\letterspace> }%
1921 }
1922 </pdfTeX-def|luatex-def|letterspace>

```

\MT@ls@fontspec@colon Add the kernfactor feature to a font loaded by fontspec (we might have to add the colon ourselves).

\MT@ls@fontspec@font

```

1923 <*\luatex-def|letterspace>
1924 \def\MT@ls@fontspec@colon#1:#2:#3:#4@nil{\ifx\\#3\\#1:#2\else#1:#2:#3\fi}
1925 \def\MT@ls@fontspec@font#1 #2@nil{%
1926   "\MT@ls@fontspec@colon#1:::\relax@nil
1927   kernfactor=\MT@minus \ifnum\MT@letterspace@=1000 1\else 0.%
1928     \ifnum\MT@minus\MT@letterspace@<100 0\fi
1929     \ifnum\MT@minus\MT@letterspace@<10 0\fi
1930     \number\MT@minus\MT@letterspace@ \fi;"
1931   \ifx\\#2\\ at \f@size pt\else#2\fi\relax
1932 }
1933 </luatex-def|letterspace>

```

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

```

1934 <*\pdfTeX-def|luatex-def>
1935 \def\MT@get@tr@opt{%
1936   \MT@set@listname
1937   \let\MT@tr@factor@\@m

```

\MT@tr@unit@ Different unit (for letterspace and/or (outer)spacing)?

```

1938 \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name @unit}{%
1939   \MT@let@cn\MT@tr@unit@{MT@tr@cc@MT@tr@cc@name @unit}%
1940   \ifdim\MT@tr@unit@=1em
1941     \let\MT@tr@unit@\undefined
1942   \else
1943     \MT@get@unit\MT@tr@unit@
1944   \fi
1945 }%
1946 \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name}{%
1947   \MT@let@cn\MT@letterspace{MT@tr@cc@MT@tr@cc@name}%
1948   \MT@ifdefined@c@T\MT@tr@unit@{%
1949     \let\@tempb\MT@letterspace
1950     \MT@scale@to@em
1951     \edef\MT@letterspace{\number\@tempcntb}%
1952   }%
1953 }%

```

\MT@tr@ispace Adjust interword spacing.

```

1954 \MT@get@tr@opt@{spacing} {ispace}%
1955 \MT@get@tr@opt@{outerspacing}{ospace}%

```

\MT@tr@okern Adjust outer kerning.

```

1956 \MT@get@tr@opt@{outerkerning}{okern}%

```

\MT@tr@ligatures Which ligatures should we disable (empty means all, undefined none)?

```

1957 \MT@get@tr@opt@{noligatures} {ligatures}%
1958 }

```

\MT@get@tr@opt@

```

1959 \def\MT@get@tr@opt@#1#2{%

```

```

1960 \MT@ifdefined@n@T{MT@tr@C@MT@tr@C@name @#1}%
1961 {\MT@let@nn{MT@tr@#2}{MT@tr@C@MT@tr@C@name @#1}}%
1962 }
1963 </pdfTeX-def|luatex-def>

\MT@set@lsfont      Redefine \font@name, which will be called a second later (in \selectfont).
1964 <*pdfTeX-def|luatex-def|letterspace>
1965 <plain>\MT@requires@latex2{
1966 \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.
                    Only \textls can be used in math mode (\lsstyle may be used inside another
                    text switch, of course). Still, we have to ensure that math fonts are set up again.
                    Setting \glb@currsiz globally to \empty (our previous solution) could throw us
                    into an infinite loop (e.g., with the psnfss packages, via \every@math@size), so
                    we issue \glb@settings instead. However, in certain situations, we may still miss
                    some math fonts, so let's try to also enforce it by emptying \glb@currsiz, fingers
                    crossed. The overhead seems small.

1967 \DeclareRobustCommand\lsstyle{%
1968 \not@math@alphabet\lsstyle\textls
1969 \let\glb@currsiz\empty
1970 <pdfTeX-def|luatex-def> \MT@maybe@gobble@with@tikz{\aftergroup\glb@settings}%
1971 <pdfTeX-def|luatex-def> \def\MT@feat{tr}%
1972 \let\MT@tracking\MT@set@tr@codes
1973 \selectfont
1974 }

```

Now the definitions for the letterspace package with plain T_EX.

```

1975 <*plain>
1976 }{
1977 \def\MT@set@lsfont{\MT@lsfont}
1978 \def\lsstyle{%
1979 \begingroup
1980 \escapechar\m@ne
1981 \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1982 \MT@set@tr@codes
1983 \endgroup
1984 }
1985 \let\textls\undefined
1986 \let\lslig\undefined
1987 }
1988 </plain>

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

1989 \DeclareRobustCommand\lslig[1]{%
1990 {\MT@ifdefined@C@TF\MT@curr@ls{%
1991 \escapechar\m@ne
1992 \MT@get@ls@basefont
1993 \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1994 \kern\MT@outer@kern
1995 \font@name #1%
1996 \kern\MT@outer@kern
1997 }{#1}}%
1998 }

```

\MT@ls@basefont pdfT_EX cannot letterspace fonts that already are letterspaced. Therefore, we have
\MT@get@ls@basefont 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, would destroy all previously set up micro-typographic features of the font.

```

1999 \def\MT@get@ls@basefont{%
2000   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
2001   \expandafter\ifx\MT@ls@basefont\relax
2002     \MT@exp@two@c\MT@glet\MT@ls@basefont\font@name
2003   \else
2004     <debug>\MT@info@n1{1}{... fixing base font}%
2005     \MT@exp@two@c\let\font@name\MT@ls@basefont
2006   \fi
2007 }

\MT@set@ls@basefont    If tracking is switched off in the middle of the document, or if \textls is called
\MT@set@tr@zero        with a zero letterspacing amount, we have to retrieve the base font and select it.

2008 \def\MT@set@ls@basefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
2009 \def\MT@set@tr@zero{%
2010   <debug>\MT@info@n1{1}{... zero tracking}%
2011   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
2012   \expandafter\ifx\MT@ls@basefont\relax \else
2013     <debug>\MT@info@n1{1}{... fixing base font}%
2014     \aftergroup\MT@set@ls@basefont
2015   \fi
2016 }
2017 </pdfTeX-def|luatex-def|letterspace>

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

2018 <*pdfTeX-def|luatex-def>
2019 <pdfTeX-def>\MT@requires@pdfTeX7{
2020   \def\MT@tr@noligatures{%
2021     \ifx\MT@tr@noligatures\@empty
2022       \MT@noligatures@\MT@lsfont\@undefined
2023     \else
2024       \MT@noligatures@\MT@lsfont\MT@tr@noligatures
2025     \fi
2026   }
2027 <*pdfTeX-def>
2028 }{
2029   \def\MT@tr@noligatures{%
2030     \MT@warning@n1{%
2031       Disabling selected ligatures is only possible since\MessageBreak
2032       pdfTeX 1.40.4. Disabling all ligatures instead}%
2033     \MT@glet\MT@tr@noligatures\relax
2034   }
2035 }
2036 </pdfTeX-def>

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

2037 \newskip\MT@outer@space

\MT@tr@set@space       Adjust interword spacing (\fontdimen 2,3,4) 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.

2038 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
2039   <debug>\MT@info@n12{... orig. space: \the\fontdimen2\MT@lsfont,
2040   <debug>   \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
2041   <debug>   \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
2042   \let\MT@temp\@empty
2043   \MT@tr@set@space@{#1}{#4}{2}\@empty
2044   \MT@tr@set@space@{#2}{#5}{3}\@plus
2045   \MT@tr@set@space@{#3}{#6}{4}\@minus
2046   \MT@glet@c\MT@outer@space\expandafter\string\font@name\MT@temp
2047   <debug>\MT@info@n12{... inner space: \the\fontdimen2\MT@lsfont,
2048   <debug>   \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
2049   <debug>\MT@info@n12{... outer space: \MT@temp}%
2050 }

```


`\MT@tr@set@space@` If settings for outer spacing (#2) don't exist, they will be inherited from the inner spacing settings (#1).

```

2051 \def\MT@tr@set@space@#1#2#3#4{%
2052   \MT@ifempty{#2}{%
2053     \MT@ifempty{#1}\relax{%
2054       \MT@tr@set@space@@{#1}{#3}{1000}%
2055       \fontdimen#3\MT@lsfont=\@tempdima
2056     }%
2057     \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
2058   }{%
2059     \MT@tr@set@space@@{#2}{#3}{2000}%
2060     \edef\MT@temp{\MT@temp#4\the\@tempdima}%
2061     \MT@ifempty{#1}\relax{%
2062       \MT@tr@set@space@@{#1}{#3}{1000}%
2063       \fontdimen#3\MT@lsfont=\@tempdima
2064     }%
2065   }%
2066 }

```

`\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.

```

2067 \def\MT@tr@set@space@@#1#2#3{%
2068   \MT@test@ast#1*\@nil{%
2069     \MT@ifdefined@c@TF\MT@tr@unit@
2070     {\edef\@tempb{#1}\MT@scale@to@em}
2071     {\@tempcntb=#1\relax}%
2072     \@tempdima=\dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax

```

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

```

2073   \ifnum#2=\tw@
2074     \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
2075   \fi
2076 }{%
2077   \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
2078   \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
2079 }%
2080 (debug)\MT@info{n13}{... : font dimen #2 (#1): \the\@tempdima}%
2081 }

```

`\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).

```

2082 \def\MT@tr@outer@l{%
2083   \ifhmode
2084     \ifdim\lastskip>5sp
2085       \edef\x{\the\lastskip minus 0pt}%
2086       \setbox\z@\hbox{\MT@outer@space=\x}%
2087       \ifdim\wd\z@>\z@
2088         (debug)\MT@info{2}{[[ adjusting pre space: \the\MT@outer@space}%
2089         \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

2090   \let\MT@ls@outer@k\relax
2091   \else

```

The ragged2e package sets `\spaceskip` without glue.

```

2092   \ifdim\lastskip=%
2093     \ifnum\spacefactor<2000
2094       \spaceskip
2095     \else
2096       \ifdim\xspaceskip=\z@
2097         \dimexpr\spaceskip+\fontdimen7\font@name\relax
2098       \else
2099         \xspaceskip

```

```

2100         \fi
2101     \fi
2102 (debug)\MT@edinfo2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
2103         \unskip \hskip\MT@outer@space\relax
2104         \let\MT@ls@outer@k\relax
2105     \fi
2106 \fi
2107 \fi
2108 \fi
2109 }

```

`\MT@tr@outer@next` microtype also adjusts spacing. 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`).

```

2110 \def\MT@tr@outer@r{%
2111     \futurelet\MT@tr@outer@next\MT@tr@outer@r@
2112 }

```

`\MT@if@outer@next` We avoid using `\ifx` tests, in case `\MT@tr@outer@next` is `\let` to `\fi` etc.

```

2113 \def\MT@if@outer@next#1{%
2114     \ifx\MT@tr@outer@next#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
2115 }

```

`\MT@tr@outer@r@`

```

2116 \def\MT@tr@outer@r@{%
2117     \def\MT@temp*{}%

```

Don't adjust in math mode. There was a tricky bug when `\textls` was the last command in a `\mathchoice` group.

```

2118     \ifmmode \else

```

A similar bug occurred when adjustment would happen inside a discretionary group, which we prevent here. This only works with e-TeX (which we know is available).

```

2119     \ifnum\currentgrouptype=10 \else
2120         \def\MT@temp*##1{\ifhmode\hskip\MT@outer@space
2121 (debug)\MT@edinfo2{[[[ adjusting post space (1): \the\MT@outer@space}%
2122         \fi}%
2123         \expandafter\ifcat\expandafter\noexpand\csname MT@tr@outer@next\endcsname\egroup
2124         \ifhmode\unkern\fi\egroup
2125         \MT@set@curr@ok \MT@set@curr@os
2126         \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=%}%
2127     \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.

```

2128     \MT@if@outer@next\maybe@ic{%
2129         \MT@set@curr@ok \MT@set@curr@os
2130         \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=%}%
2131     }{%

```

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).

```

2132     \MT@if@outer@next\check@icr{%
2133         \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\MT@temp=%}%
2134     }{%
2135         \MT@if@outer@next\@sptoken{%
2136             \def\MT@temp* {\ifhmode\hskip\MT@outer@space
2137 (debug)\MT@edinfo2{[[[ adjusting post space (2): \the\MT@outer@space}%

```

xspace requires special treatment.

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

\MT@tr@outer@icr Helper macros for the italic correction mess.

<code>\MT@xspace</code>	If the group is followed by <code>\xspace</code> , we first feed <code>\xspace</code> with the next token, then
<code>\MT@xspace@</code>	check whether it has inserted a space. <code>\@let@token</code> might be something evil, so it should be encapsulated here.

For older pdfT_EX versions and LuaT_EX, throw an error.

And for $X_{\neg T_F} X$, too.

```

2182 \pdfTeX-def|luatex-def)
2183 {xetex-def)
2184 \DeclareRobustCommand\lssstyle{%
2185   \MT@error{Letterspacing currently doesn't work with xetex}
2186   {Run pdfTeX or luatex, or use the 'soul' package instead.}%
2187   \MT@glet\lssstyle\relax
2188 }

```

```

2189 </xetex-def>
\textls      This command may be used like the other text commands. The starred version
\MT@ls@adjust@ removes kerning on the sides. The optional argument changes the letterspacing
               factor.
2190 <*package|letterspace>
2191 \DeclareRobustCommand\textls{%
2192   \ifstar{\let\MT@ls@adjust@MT@ls@adjust@empty\MT@textls}%
2193   {\let\MT@ls@adjust@MT@ls@adjust@relax\MT@textls}%
2194 }

\MT@textls   This is now almost LATEX's \DeclareTextFontCommand, with the difference that we
\MT@letterspace@ adjust the outer spacing and kerning also for \lsstyle, while LATEX's text switches
               don't bother about italic correction.
2195 \newcommand\MT@textls[2][]{%
2196   \ifmmode
2197     \nfss@text{\MT@ls@set@ls{#1}\lsstyle#2}%
2198   \else
2199     \hmode@bgroup
2200     \MT@ls@set@ls{#1}%
2201     \lsstyle #2%
2202     \expandafter
2203     \egroup
2204   \fi
2205 }

\MT@ls@adjust Set current letterspacing amount and outer kerning. This has to be done inside the
\MT@ls@adjust@empty same group as the letterspacing command.
\MT@ls@adjust@relax 2206 \def\MT@ls@adjust@empty{\let\MT@ls@adjust@empty}
\MT@ls@set@ls      2207 \def\MT@ls@adjust@relax{\let\MT@ls@adjust@relax}
2208 \def\MT@ls@set@ls#1{%
2209   \MT@ifempty{#1}%
2210   {\let\MT@letterspace@ \@undefined}%
2211   {\KV@sp@def\MT@letterspace@{#1}%
2212     \edef\MT@letterspace@{\number\MT@letterspace@}%
2213     \MT@ls@too@large\MT@letterspace@}%
2214   \MT@ls@adjust@
2215 }

\MT@ls@too@large Test whether letterspacing amount is too large.
2216 \def\MT@ls@too@large#1{%
2217   \ifnum#1>\MT@tr@max
2218     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
2219     \let#1\MT@tr@max
2220   \else
2221     \ifnum#1<\MT@tr@min
2222       \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
2223       \let#1\MT@tr@min
2224     \fi
2225   \fi
2226 }

\MT@outer@kern This dimen is used for the starred version of \textls, for \lslig and for adjusted
\MT@tr@set@okern outer kerning.
2227 \newdimen\MT@outer@kern
2228 </package|letterspace>
2229 <*pdfTeX-def|LaTeX-def>
2230 \def\MT@tr@set@okern#1,#2,{%
2231   \let\MT@temp@empty
2232   \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
2233   \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
2234   \MT@gl@et@nc{\MT@outer@kern\expandafter\string\font@name}\MT@temp
2235 <debug>\MT@din@fo@n12{... outer kerning: (#1,#2)}

```

```

2236 <debug>                = \nameuse{MT@outer@kern\expandafter\string\font@name}}%
2237 }

\MT@tr@set@okern@

2238 \def\MT@tr@set@okern@#1{%
2239   \MT@test@ast#1*\nil{%
2240     \MT@ifdefined@ec@TF\MT@tr@unit@
2241     {\edef\@tempb{#1}\MT@scale@to@em}
2242     {\@tempcntb=#1\relax}%
2243     \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
2244   }{%
2245     \MT@ifempty\@tempa{\let\@tempa\@m}\relax
2246     \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
2247               * \fontdimen6\MT@lsfont/2000\relax
2248   }%
2249   \advance\@tempdima -\dimexpr \MT@letterspace@ sp
2250               * \fontdimen6\MT@lsfont/2000\relax
2251   \edef\MT@temp{\MT@temp{\the\@tempdima}}%
2252 }
2253 </pdfTeX-def|luaTeX-def>

\MT@ls@outer@k Adjust outer kerning. We additionally add a marker (\kern3sp\kern-3sp) for cases
of nested letterspacing without anything actually printed.

2254 <pdfTeX-def|luaTeX-def|letterspace>
2255 \def\MT@ls@outer@k{%
2256   \ifhmode
2257     \ifdim\lastkern=-3sp \unkern
2258     \ifdim\lastkern=3sp \kern-3sp
2259     \expandafter\expandafter\expandafter\@gobble
2260     \else \unkern
2261     \expandafter\expandafter\expandafter\@firstofone
2262     \fi
2263     \else
2264     \expandafter\@firstofone
2265     \fi
2266     {\kern\MT@outer@kern\kern3sp\kern-3sp\relax}%
2267   \fi
2268 }
2269 </pdfTeX-def|luaTeX-def|letterspace>

```

14.2.6 Disabling ligatures

\MT@noligatures The possibility to disable ligatures is a new features of pdfTeX 1.30, and also works with LuaTeX.

```

2270 <pdfTeX-def|luaTeX-def>
2271 <pdfTeX-def>\MT@requires@pdfTeX5{
2272 \def\MT@noligatures{%
2273   \MT@dotrue
2274   \let\@tempa\MT@n@setname
2275   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
2276     \MT@ifdefined@n@TF{MT@checklist@##1}%
2277     {\csname MT@checklist@##1\endcsname}%
2278     {\MT@checklist@{##1}}}%
2279   {nl}}%
2280 }%
2281 \ifMT@do
2282   \MT@noligatures@MT@font\MT@n@ligatures
2283 \fi
2284 }

```

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

```

2285 <luaTeX-def>\MT@requires@luaTeX4{\let\pdfnoligatures\ignoreligaturesinfont}\relax
2286 \def\MT@noligatures@#1#2{%

```

```
2287 \MT@ifdefined@ec@TF#2{%
```

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

```
2288 \MT@ifdefined@ec@TF\tagcode{%
```

No 'inputenc' key.

```
2289 \let\MT@warn@maybe@inputenc@empty
2290 \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
2291 \MT@map@clist@c#2{%
2292   \KV@sp@def\@tempa{##1}\MT@get@slot
2293   \ifnum\MT@char>\m@ne
2294     \tagcode#1\MT@char=\m@ne
```

With LuaTeX, we additionally register the ligatures that should be inhibited in a table (used by the luaotfload function keepligature).

```
2295 <luatex-def> \MT@if@luaotf@font
2296 <luatex-def> {\MT@lua{microtype.noligatures([[#1]],[[\MT@char]])}}\relax
2297 \fi
2298 }%
2299 \MT@vinfo{... Disabling ligatures for characters: #2}%
2300 }{%
2301   \pdfnoligatures#1%
2302   \MT@warning{Cannot disable selected ligatures (pdfTeX doesn't\MessageBreak
2303     know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
2304     the font instead}%
2305 }%
2306 }{%
2307   \pdfnoligatures#1%
2308 <luatex-def> \MT@if@luaotf@font
2309 <luatex-def> {\MT@lua{microtype.noligatures([[#1]],"_all_")}}\relax
2310 \MT@vinfo{... Disabling all ligatures}%
2311 }%
2312 }
2313 <pdfTeX-def>\relax
2314 </pdfTeX-def|luatex-def>
```

For each potential ligature, luaotfload will call the keepligature function, which expects the first node of the ligature, to check whether they should be kept or inhibited. Here's our concoction of this function. The table microtype.ligs will be populated in \MT@noligatures@.

```
2315 <*luafile>
2316 microtype.ligs = microtype.ligs or { }
2317
2318 local function noligatures(fontcs,liga)
2319   local fontcs = match(fontcs,"([^\ ]+)"
2320   microtype.ligs[fontcs] = microtype.ligs[fontcs] or { }
2321   table.insert(microtype.ligs[fontcs],liga)
2322 end
2323 microtype.noligatures = noligatures
2324
2325 local function keepligature(c)
2326   local nodedirect = node.direct
2327   local getfield = nodedirect.getfield
2328   local getfont = nodedirect.getfont
2329   local f,ch
2330   if type(c) == "userdata" then -- in older luaotfload versions, c was a node
2331     f = c.font
2332     ch = c.components.char
2333   else -- since 2.6, c is a (direct node) number
2334     f = getfont(c)
2335     ch = getfield(getfield(c,"components"),"char")
2336   end
2337   -- if ch then -- should always be true
2338   local lig = microtype.ligs[match(tex.fontidentifier(f),"\\([^\ ]+)")]
```

```

2339 if lig then
2340   for _,lig in pairs(ligs) do
2341     if lig == "_all_" or tonumber(lig) == ch then
2342       return false
2343     end
2344   end
2345 end
2346 return true
2347 -- end
2348 end
2349
2350 if luaotfload and luaotfload.letterspace then
2351   if luaotfload.letterspace.keepligature then
2352     microtype.info("overwriting function `keepligature'")
2353   end
2354   luaotfload.letterspace.keepligature = keepligature
2355 end
2356
2357 (luafile)

```

14.2.7 Loading the configuration

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

```

2358 (*package)
2359 \def\MT@load@list#1{%
2360   \edef\@tempa{#1}%
2361   \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa @load}%
2362   \MT@ifstreq\@tempa\@tempb{%
2363     \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempa' cannot load itself}{}%
2364   }%
2365   \ifx\@tempb\relax \else
2366     \MT@ifdefinedn@TF{MT@MT@feat @c@\@tempb}{%
2367       \MT@vinfo{... : First loading \@nameuse{MT@abbr@MT@feat} list `@\@tempb'}%
2368       \begingroup
2369         \MT@load@list\@tempb
2370       \endgroup
2371       \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list
2372         \noexpand\MessageBreak`@\@tempb'}%
2373       \MT@let@cn\@tempc{MT@MT@feat @c@\@tempb}%
2374       \expandafter\MT@set@codes\@tempc,\relax,%
2375     }{%
2376       \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempb' undefined.\MessageBreak
2377         Cannot load it from list `@\@tempa'}{%
2378     }%
2379   \fi
2380 }%
2381 }

```

`\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.

```

2382 \let\MT@file@list\empty
2383 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

2384 \MT@in@clist{#1}\MT@file@list
2385 \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.

```

2386 \MT@begin@catcodes
2387 \let\MT@begin@catcodes\relax
2388 \let\MT@end@catcodes\relax
2389 \InputIfFileExists{mt-#1.cfg}{%

```

```

2390     \edef\MT@curr@file{mt-#1.cfg}%
2391     \MT@vinfo{... Loading configuration file \MT@curr@file}%
2392     \MT@xadd\MT@file@list{#1,}%
2393   }{%
2394     \MT@get@basefamily#1\@empty\@empty\@empty\@nil
2395     \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2396     \ifMT@inlist@
2397       \MT@xadd\MT@file@list{#1,}%
2398     \else
2399       \InputIfFileExists{mt-\@tempa.cfg}{%
2400         \edef\MT@curr@file{mt-\@tempa.cfg}%
2401         \MT@vinfo{... Loading configuration file \MT@curr@file}%
2402         \MT@xadd\MT@file@list{\@tempa,#1,}%
2403       }{%
2404         \MT@vinfo{... No configuration file mt-#1.cfg}%
2405         \MT@xadd\MT@file@list{#1,}%
2406       }%
2407     \fi
2408   }%
2409 \endgroup
2410 \fi
2411 }

```

`\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.

```

2412 \def\MT@cfg@catcodes{%
2413   \makeatletter
2414   \catcode`\^7%
2415   \catcode`\_9%
2416   \catcode`\^^I9%
2417   \catcode`\^^M9%
2418   \catcode`\z@
2419   \catcode`\{\@one
2420   \catcode`\}\@tw@
2421   \catcode`\#6%
2422   \catcode`\%14%
2423   \MT@map@tlist@n
2424   {!\\"$&'(\)*+,\-\.\/:;<=>?[\]\_~\|^-}%
2425   \@makeother
2426 }

```

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

```

2427 \def\MT@begin@catcodes{%
2428   \begingroup
2429   \MT@cfg@catcodes
2430 }

```

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

```

2431 \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.

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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```

2432 \def\MT@get@basefamily#1#2#3#4\@nil{%
2433   \ifx\@empty#4%
2434     \def\@tempa{#1#2#3}%
2435   \else
2436     \let\@tempa\@empty
2437     \edef\@tempb{#1#2#3#4}%
2438     \expandafter\MT@get@basefamily@\@tempb\@nil
2439   \fi
2440 }

```

\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 ‘pplx’ would be truncated to ‘p’.

```

2441 \def\MT@get@basefamily@#1#2\@nil{%
2442   \edef\@tempa{\@tempa#1}%
2443   \ifx\#2\@expandafter\@gobble\else\expandafter\@firstofone\fi
2444   {\MT@in@tlist{#2}\MT@variants
2445     \ifMT@inlist\else\MT@get@basefamily@#2\@nil\fi}%
2446 }

```

\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@ 2447 \def\MT@get@listname#1{%
2448   (debug)\MT@edinfo@n1{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
2449   \let\MT@listname\@undefined
2450   \def\@tempb{#1}%
2451   \MT@map@tlist@c\MT@try@order\MT@get@listname@
2452 }
2453 \def\MT@get@listname@#1{%
2454   \expandafter\MT@next@listname#1%
2455   \ifx\MT@listname\@undefined \else
2456     \expandafter\MT@tlist@break
2457   \fi
2458 }

```

\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.

```

2459 \def\MT@try@order{%
2460   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2461   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2462 }

```

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

```

2463 \def\MT@next@listname#1#2#3#4{%
2464   \ifnum#1=\z@\MT@nofamilytrue\fi
2465   \edef\@tempa{\MT@encoding
2466     /\ifnum#1=\@one \MT@family \fi
2467     /\ifnum#2=\@one \MT@series \fi
2468     /\ifnum#3=\@one \MT@shape \fi
2469     /\ifnum#4=\@one *\fi
2470   \MT@context}%

```

```

2471 (debug)\MT@info@n1{1}{trying \@tempa}%
2472 \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
2473   \MT@next@listname@#4%
2474 }{%

```

Also try with an alias family.

```

2475 \ifnum#1=\@ne
2476   \ifx\MT@familyalias\@empty \else
2477     \edef\@tempa{\MT@encoding
2478       /\MT@familyalias
2479       /\ifnum#2=\@ne \MT@series\fi
2480       /\ifnum#3=\@ne \MT@shape\fi
2481       /\ifnum#4=\@ne *\fi
2482       \MT@context}%
2483 (debug)\MT@info@n1{1}{(alias) \@tempa}%
2484 \MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%
2485   \MT@next@listname@#4%
2486 }{%
2487   \fi
2488   \fi
2489 }%
2490 }

```

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

```

2491 \def\MT@next@listname@#1{%
2492   \ifnum#1=\@ne
2493     \MT@exp@cs\MT@in@rlist{MT@\@tempb @\@tempa @sizes}%
2494     \ifMT@inlist@
2495       \let\MT@listname\MT@size@name
2496     \fi
2497   \else
2498     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
2499   \fi
2500 }

```

\MT@if@list@exists

```

\MT@context 2501 \def\MT@if@list@exists{%
2502   \MT@let@cn\MT@context{MT@\MT@feat @context}%
2503   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
2504   \MT@get@listname{\MT@feat @c}%
2505   \MT@ifdefined@c@TF\MT@listname{%
2506     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
2507     \ifMT@nonselected
2508       \MT@vinfo{... Applying non-selected expansion (list `\'MT@listname')}%
2509     \else
2510       \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `\'MT@listname'}%
2511     \fi
2512     \@firstoftwo
2513   }{%

```

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

```

2514   \MT@let@cn{MT@\MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

2515   \ifMT@nonselected
2516     \MT@vinfo{... Applying non-selected expansion (no list)}%
2517   \else

```

Tracking doesn't require a list, either.

```

2518   \MT@ifstreq\MT@feat{tr}\relax{%
2519     \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
2520       for font\MessageBreak`\'MT@@font'%
2521     \ifx\MT@context\@empty\else\space(context: `\'MT@context')\fi.
2522     Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%

```

```

2523     }%
2524     \fi
2525     \@secondoftwo
2526   }%
2527 }

```

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

```

\MT@context 2528 \def\MT@get@inh@list{%
2529   \let\MT@context\@empty

2530   \MT@get@listname{\MT@feat @inh}%
2531   \MT@ifdefined@c@TF\MT@listname{%
2532     \MT@edefn{MT@\MT@feat @inh@name}{\MT@listname}%
2533   }%
2534   \MT@edefn{MT@\MT@feat @inh@name}{\MT@listname}%
2535   \MT@let@cn\@tempc{MT@\MT@feat @inh@MT@listname}%

```

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

```

2536   \ifx\@tempc\@empty \else
2537   \debug\MT@info@n1{1}{parsing inheritance list ...}%

```

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

```

2538   \begingroup
2539   \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak` \MT@listname'}%
2540   \MT@set@inputenc{inh}%
2541   \expandafter\MT@inh@do\@tempc,\relax,%
2542   \MT@gl@et@nc{MT@\MT@feat @inh@MT@listname}\@empty
2543   \endgroup
2544   \fi
2545   }%
2546   \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
2547   }%
2548 }

```

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@ 2549 \def\MT@get@slot{%
2550   \escapechar`\\
2551   \let\MT@char@m@ne
2552   \MT@noresttrue

```

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

```

2553   \MT@toks=\expandafter{\@tempa}%

```

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.

```

2554   \MT@exp@two@c\MT@is@active\string\@tempa\@nil

```

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

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

```

2555   \expandafter\MT@is@letter\@tempa\relax\relax
2556   \ifnum\MT@char@ < \z@

```

- 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.

```
2557 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
2558 \MT@is@symbol
```

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

```
2559 {\expandafter\MT@is@composite\@tempa\relax\relax}%
2560 \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.

```
2561 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2562 \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2563 \fi
2564 \fi

2565 \let\MT@char\MT@char@
2566 \MT@get@slot@
2567 \escapechar\m@ne
2568 }
2569 /package
```

$\backslash MT@get@slot@$

```
2570 <pdfTeX-def|luatex-def|xetex-def>
2571 \def\MT@get@slot@{%
```

If it's a legacy (i.e., TFM) font, proceed as usual.

```
2572 <xetex-def> \ifnum\XeTeXfonttype\MT@font=\z@
2573 \ifnum\MT@char > \m@ne
```

In Lua_TE_X, it may also be a glyph name, prefixed with \backslash' .

```
2574 <luatex-def>
2575 \ifnum\MT@char=47\relax
2576 \ifMT@noreset \else
2577 \@tempcnta=\MT@lua{
2578   local glyph = microtype.name_to_slot([[ \expandafter\@gobble\@tempa ]],true)
2579   if glyph then tex.write(glyph)
2580   else tex.write(-1)
2581   end
2582 } \relax
2583 \ifnum\@tempcnta<\z@
2584 \MT@warn@unknown
2585 \let\MT@char\m@ne
2586 \else
2587 \edef\MT@char{\the\@tempcnta}%
2588 <debug> \MT@edinfo@n1{3}{> ` \the\MT@toks' is a glyph name (\the\@tempcnta)}%
2589 \fi
2590 \fi
2591 \else
2592 /luatex-def
```

If the user has specified something like $\backslash'fi$, 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.

```
2593 \ifMT@noreset \else
```

```

2594     \MT@warn@rest
2595 <pdfTeX-def|luatex-def> \let\MT@char\m@ne
2596 <xetex-def> \let\MT@char\@empty
2597     \fi
2598 <luatex-def> \fi
2599     \else
2600     \MT@warn@unknown
2601 <xetex-def> \let\MT@char\@empty
2602     \fi
2603 <*xetex-def>
2604     \else

```

There are more possibilities for X_YTeX: It may also be a glyph name (prefixed with ‘/’). We indicate this to \MT@get@charwd by reversing the sign of \MT@char@.

```

2605     \ifnum\MT@char=47\relax
2606     \ifMT@norest \edef\MT@char{U47}%
2607     \else
2608     \@tempcnta=XeTeXglyphindex"\expandafter\@gobble\@tempa"\relax
2609     \ifnum\@tempcnta=z@
2610     \MT@warn@unknown
2611     \let\MT@char\@empty
2612     \else
2613     \edef\MT@char{\@tempa\space}%
2614     \edef\MT@char@{-\the\@tempcnta}%
2615 <debug>\MT@info@nl{3}{> `the\MT@toks' is a glyph name (\the\@tempcnta)}%
2616     \fi
2617     \fi
2618     \else
2619     \ifnum\MT@char > \m@ne
2620     \ifMT@norest

```

Or, it’s a Unicode number, which we mustn’t translate into a glyph number, since the latter is font-specific.

```

2621     \@tempcnta=XeTeXcharglyph\MT@char\relax
2622     \ifnum\@tempcnta=z@
2623     \MT@info@missing@char
2624     \let\MT@char\@empty
2625     \else
2626 <debug>\MT@info@nl{3}{> (glyph number: \the\@tempcnta,
2627 <debug> glyph name: \XeTeXglyphname\MT@font\@tempcnta)}%
2628     \edef\MT@char{U\MT@char}%
2629     \fi
2630     \else
2631     \MT@warn@rest
2632     \let\MT@char\@empty
2633     \fi
2634     \else
2635     \MT@warn@unknown
2636     \let\MT@char\@empty
2637     \fi
2638     \fi
2639     \fi
2640 </xetex-def>
2641 }
2642 </pdfTeX-def|luatex-def|xetex-def>

```

This is the lua function to translate glyph name into slot number. Beginning with v2.2, luaotfload provides this function in an API, which we use if available, but (for now, at least) keep the old code for backward compatibility.

```

2643 <*luafile>
2644 if luaotfload and luaotfload.aux and luaotfload.aux.slot_of_name then
2645     local slot_of_name = luaotfload.aux.slot_of_name
2646     microtype.name_to_slot = function(name, unsafe)
2647         return slot_of_name(font.current(), name, unsafe)

```

```

2648 end
2649 else
2650   -- we dig into internal structure (should be avoided)
2651   local function name_to_slot(name, unsafe)
2652     if fonts then
2653       local unicodes
2654       if fonts.ids then -- legacy luaotfload
2655         local tfmdata = fonts.ids[font.current()]
2656         if not tfmdata then return end
2657         unicodes = tfmdata.shared.otfdata.luaotfload.unicodes
2658       else -- new location
2659         local tfmdata = fonts.hashes.identifiers[font.current()]
2660         if not tfmdata then return end
2661         unicodes = tfmdata.resources.unicodes
2662       end
2663       local unicode = unicodes[name]
2664       if unicode then -- does the 'or' branch actually exist?
2665         return type(unicode) == "number" and unicode or unicode[1]
2666       end
2667     end
2668   end
2669   microtype.name_to_slot = name_to_slot
2670 end
2671
2672 </luafile>

```

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

\MT@max@char Warning if resulting character or slot number is too large.

\MT@max@slot 2673 *<*pdfTeX-def|luaTeX-def|xetex-def>*
2674 \def\MT@max@char
2675 *<pdfTeX-def>* {127 }
2676 *<luaTeX-def|xetex-def>* {1114111 }
2677 \def\MT@max@slot
2678 *<pdfTeX-def>* {255 }
2679 *<luaTeX-def|xetex-def>* {1114111 }
2680 *</pdfTeX-def|luaTeX-def|xetex-def>*

\ifMT@noest Test whether all of the string has been used up.

```

2681 <*package>
2682 \newif\ifMT@noest

2683 \def\MT@is@letter#1#2\relax{%
2684   \ifcat a\noexpand#1\relax
2685     \edef\MT@char@{\number`#1}%
2686     \ifx\#2\%
2687       <debug>\MT@info@n{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2688       \else
2689         \MT@noestfalse
2690       \fi
2691     \else
2692       \ifcat !\noexpand#1\relax
2693         \edef\MT@char@{\number`#1}%
2694       <debug>\MT@info@n{3}{> `the\MT@toks' is a character (\MT@char@)}%
2695       \ifx\#2\%
2696         \ifnum\MT@char@ > \MT@max@char \MT@warn@ascii \fi
2697       \else
2698         \MT@noestfalse
2699         \expandafter\MT@is@number#1#2\relax\relax
2700       \fi
2701     \fi
2702   \fi
2703 }

```

\MT@is@number Numbers may be specified as a three-digit decimal number (029), as a hexadecimal number (prefixed with "0x": "1D") or as a octal number (prefixed with "0o": "35"). They

must consist of at least three characters (including the prefix), that is, "F is not permitted.

```

2704 \def\MT@is@number#1#2#3\relax{%
2705   \ifx\relax#3\relax \else
2706     \ifx\relax#2\relax \else
2707       \MT@noesttrue
2708       \if#1"\relax
2709         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2710 <debug>\MT@info@n1{3}> ... a hexadecimal number: \MT@char@}%
2711       \else
2712         \if#1'\relax
2713         \def\MT@char@{\number#1#2#3}%
2714 <debug>\MT@info@n1{3}> ... an octal number: \MT@char@}%
2715       \else
2716         \MT@ifint{#1#2#3}{%
2717           \def\MT@char@{\number#1#2#3}%
2718 <debug>\MT@info@n1{3}> ... a decimal number: \MT@char@}%
2719         } \MT@noestfalse
2720       \fi
2721     \fi
2722     \ifnum\MT@char@ > \MT@max@slot
2723       \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2724       \let\MT@char@\m@ne
2725     \fi
2726   \fi
2727 \fi
2728 }
```

`\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.

```

2729 \def\MT@is@active#1#2\@nil{%
2730   \ifnum\catcode`#1 = \active
2731     \begin@group
2732     \set@display@protect
2733     \let\IeC\@firstofone
2734     \let\@in@penc@undefined@\MT@undefined@char
```

Unicode handling has changed again with L^AT_EX 2019/10/01.

```

2735   \let\UTF@two@octets@noexpand\@empty
2736   \let\UTF@three@octets@noexpand\@empty
2737   \let\UTF@four@octets@noexpand\@empty
```

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

```

2738   \def\UTFviii@defined##1{\ifx ##1\relax
2739     \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
```

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

```

2740   \MT@ifdefined@c@T\PrerenderUnicode
2741     {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
```

The `\expandafter` hocus-pocus should please `newunicodechar`.

```

2742   \edef\x{\endgroup
2743     \def\noexpand\@tempa{\expandafter\expandafter\expandafter\@empty\@tempa}%
```

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

```

2744      \MT@toks={\the\MT@toks\space(=
2745                  \expandafter\expandafter\expandafter\@empty\@tempa)}%
2746      }%
2747      \x
2748      \fi
2749  }

```

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

```

2750 \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.

```

2751 \def\MT@is@symbol{%
2752   \expandafter\def\expandafter\MT@char\expandafter
2753     {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2754   \expandafter\expandafter\expandafter
2755     \MT@is@opt@char\MT@char\iffontchar\char\else\fi\relax
2756   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2757     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2758   \ifnum\MT@char@ < \z@

```

Since recently, some glyphs are defined optionally in L^AT_EX by checking if the glyph actually exists in the font (e.g., `\textasteriskcentered`).

```

2754   \expandafter\expandafter\expandafter
2755     \MT@is@opt@char\MT@char\iffontchar\char\else\fi\relax
2756   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2757     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2758   \ifnum\MT@char@ < \z@

```

In TU encoding, some commands (currently, `\textquotesingle`, `\textasciigrave` and `\textquotedbl`) are defined by means of the auxiliary macro `\remove@tlig`, which we take care of here.

```

2759   \expandafter\expandafter\expandafter\MT@is@tlig\MT@char\relax\relax
2760   \ifnum\MT@char@ < \z@

```

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

```

2761   \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2762   \fi
2763   \fi
2764 }

```

`\MT@is@opt@char` This seems adventurous, but we're only redefining the text command within the scope of our setup.

```

2765 \def\MT@is@opt@char#1\iffontchar#2\char#3\else#4\fi\relax{%
2766   \ifx\#1\%
2767     \iffontchar#2%
2768     \expandafter\chardef
2769       \csname\MT@encoding\MT@detokenize@c\@tempa\endcsname=#3\relax
2770     \fi
2771     \fi
2772 }

```

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

```

\MT@charstring 2773 \begingroup
2774   \catcode`\=/\z@
2775   /MT@map@tlist@n{/CHARLEX}/@makeoother
2776   /lowercase{%
2777     /def/x{/endgroup
2778     /def/MT@charstring{\CHAR}%
2779     /def/MT@is@char#1\CHAR"##2##3##4/relax{%
2780       /ifx/relax##4/relax
2781       /ifMT@xunicode
2782       /expandafter/MT@is@charx/MT@strip@prefix##1>/relax\CHAR "%
2783       /relax/relax/relax/relax/relax

```



```

2784         /fi
2785     /else
2786         /ifx/relax##1/relax
2787         /if##3\relax
2788         /edef/MT@char@{/number"##2}%
2789         /MT@ifstreq/MT@charstring{##3##4}/relax/MT@noestfalse
2790     /else
2791         /edef/MT@char@{/number"##2##3}%
2792         /MT@ifstreq/MT@charstring{##4}/relax
2793         {/MT@is@xchar##2##3|##4\CHAR"/relax}%
2794     /fi
2795 (debug) /MT@info@n1{3}{> `the/MT@toks' is a \char (/MT@char@)}%
2796     /fi
2797 /fi
2798 }%
```

`\MT@is@xchar` With fontspec's TU encoding, glyph numbers may be up to four digits.

```

2799     /def/MT@is@xchar##1|##2\CHAR"##3##4/relax{%
2800     /MT@ifstreq/MT@charstring{##3##4}%
2801     {/edef/MT@char@{/number"##1##2}}/MT@noestfalse
2802 }%
```

`\MT@charxstring` For xunicode, which doesn't `\countdef`, but rather `\defs` the chars.

```

\MT@strip@prefix 2803     /def/MT@charxstring{\CHAR "%
\MT@is@charx      2804     /def/MT@strip@prefix##1>##2/relax{##2}%
2805     /def/MT@is@charx##1\CHAR "##2##3##4##5##6/relax{%
2806     /ifx/relax##1/relax
2807     /ifx/relax##6/relax/else
2808     /edef/MT@char@{/number"##2##3##4##5}%
2809     /MT@ifstreq{\RELAX >\CHAR "}{##6}/relax/MT@noestfalse
2810 (debug) /MT@info@n1{3}{> `the/MT@toks' is a xunicode \char (/MT@char@)}%
2811     /fi
2812     /fi
2813 }%
2814 }%
2815 }
2816 /x
```

`\MT@is@tlig` This might have to change again with the next L^AT_EX release, ... or so I feared, but it still seems to be fine.

```

2817 \def\MT@is@tlig#1#2\relax{%
2818     \ifx\remove@tlig#1%
2819 (debug) \MT@info@n1{3}{> `the/MT@toks' (removing remove@tlig)}%
2820     \MT@remove@tlig
2821     \fi
2822 }
```

`\MT@remove@tlig` We remove the `\remove@tlig` command and only pass on the number.

```

2823 \def\MT@remove@tlig{%
2824     \expandafter\MT@exp@two@c\expandafter\MT@is@number
2825     \expandafter\@secondoftwo\MT@char\relax\relax
2826 }
```

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

```

2827 \def\MT@is@composite#1#2\relax{%
2828     \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`.

```

2829     \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
```

```

2830 \string\csname\MT@encoding\endcsname
2831 \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%

```

In 2017, L^AT_EX introduced a new way of declaring accented Unicode commands (`\DeclareUnicodeComposite`), which we take care of here (`\UnicodeEncodingName` has been introduced at the same time):

```

2832 \ifx\UnicodeEncodingName\undefined\else
2833 \expandafter\expandafter\expandafter
2834 \MT@is@uni@comp\MT@char\iffontchar\else\fi\relax
2835 \fi
2836 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax

```

Again, xunicode.

```

2837 \ifnum\MT@char@ < \z@
2838 \ifMT@xunicode
2839 \edef\MT@char{\MT@exp@two@c\MT@strip@prefix\meaning\MT@char>\relax}%
2840 \expandafter\MT@exp@two@c\expandafter\MT@is@charx\expandafter
2841 \MT@char\MT@charxstring\relax\relax\relax\relax\relax
2842 \fi
2843 \fi
2844 \fi
2845 }

```

`\MT@is@uni@comp` Helper for `\DeclareUnicodeComposite`.

```

2846 \def\MT@is@uni@comp#1\iffontchar#2\else#3\fi\relax{%
2847 \ifx\#1\\edef\MT@char{\iffontchar#2\fi}\fi
2848 }

```

[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@list@name 2849 \def\MT@set@list@name{%
2850 \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
2851 \~\@nameuse{MT@\MT@feat @c@name}}}%
2852 }

```

`\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.

```

2853 \def\MT@warn@ascii{%
2854 \MT@warning@n1{Character `the\MT@toks' (= \MT@char@)
2855 is outside of ASCII range.\MessageBreak
2856 You must load the `inputenc' package before using\MessageBreak
2857 8-bit characters in \MT@curr@list@name}%
2858 }

```

```

\MT@warn@number@too@large    Number too large.
2859 \def\MT@warn@number@too@large#1{%
2860   \MT@warning@nl{%
2861     Number #1 in encoding '\MT@encoding' too large!\MessageBreak
2862     Ignoring it in \MT@curr@list@name}%
2863 }

\MT@warn@rest                Not all of the string has been parsed.
2864 \def\MT@warn@rest{%
2865   \MT@warning@nl{%
2866     Unknown slot number of character!\MessageBreak`the\MT@toks'
2867     \MT@warn@maybe@inputenc!\MessageBreak
2868     in font encoding '\MT@encoding'.!\MessageBreak
2869     Make sure it's a single character!\MessageBreak
2870     (or a number) in \MT@curr@list@name}%
2871 }

\MT@warn@unknown            No idea what went wrong.
2872 \def\MT@warn@unknown{%
2873   \MT@warning@nl{%
2874     Unknown slot number of character!\MessageBreak`the\MT@toks'
2875     \MT@warn@maybe@inputenc!\MessageBreak
2876     in font encoding '\MT@encoding' in \MT@curr@list@name}%
2877 }

\MT@warn@maybe@inputenc    In case an input encoding had been requested.
2878 \def\MT@warn@maybe@inputenc{%
2879   \MT@ifdefined@n@T
2880   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
2881   { (input encoding '\@nameuse
2882     {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2883 }

```

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, it is not necessary to declare in advance which fonts 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`, `pifont` (loaded by `hyperref`), `tifa`, and probably many more. Furthermore, we had to include a hack for the `IEEEtran` class which loads all fonts in the class file itself (to fine tune inter-word spacing), and the `memoir` class, too. 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 2884 \let\MT@font@list\@empty
2885 \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`.

```
2886 \<package>
2887 \<*package|letterspace>
2888 \<plain>\MT@requires@latex2{
2889 \MT@addto@setup{%
```

`\MT@orig@pickupfont`

The `luatexja` package redefines `\char`, which will upset our parsing of text symbols and commands; instead of fixing this, we won't bother, at least for the moment, but simply issue a warning and disable all further warnings. The fix is left to the user by not specifying any text commands but only (Unicode) letters. The `xeCJK` package, or rather its `xunicode-addon`, also modifies the way text symbols are defined (like `luatexja` but in a different way). Again, we only issue a warning.

```
2890 \<package> \MT@with@package@T{luatexja}{\MT@warn@unknown@once{luatexja}}%
2891 \<package> \MT@with@package@T{xeCJK} {\MT@warn@unknown@once{xeCJK}}%
```

`microtype` also works with CJK in the sense that nothing will break when both packages are used at the same time. However, since CJK has its own way of encoding, it is currently not possible to create character-specific settings. That is, the only feature available with CJK fonts is (non-selected) expansion. (Tracking doesn't really work for other reasons.) Like us, CJK redefines `\pickup@font`.

```
2892 \@ifpackageloaded{CJK}{%
```

The `xeCJK` package in turn pretends that CJK was loaded, but does not change the definition of `\pickup@font`. With `xeCJK`, protrusion should be possible also for C/J/K characters; I haven't tried it, though.

```
2893 \@ifpackageloaded{xeCJK}{\@firstofone}{%
2894 \@ifpackagelater{CJK}{2006/10/17}% 4.7.0
2895 {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}}}%
2896 {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}%
2897 \g@addto@macro\MT@orig@pickupfont
2898 {\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).

```
2899 \@ifpackageloaded{CJKutf8}%
2900 {\@ifpackagelater{CJKutf8}{2008/05/22}% 4.8.0
2901 {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}%
2902 {\@firstoftwo}}%
2903 {\@firstoftwo}%
2904 {\g@addto@macro\MT@orig@pickupfont{%
2905 {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
```

```

2906         \define@newfont\else\xdef\font@name{%
2907             \csname \curr@fontshape/\f@size/\CJK@plane\endcsname\fi}}}%
2908     {\g@addto@macro\MT@orig@pickupfont{%
2909         {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2910             \define@newfont\def\CJK@temp{v}%
2911             \ifx\CJK@temp\CJK@plane
2912                 \expandafter\ifx\csname CJK@cmap@\f@family\CJK@plane\endcsname\relax
2913                 \else\csname CJK@cmap@\f@family\CJK@plane\endcsname\fi
2914                 \else \CJK@addcmap\CJK@plane \fi
2915             \else\xdef\font@name{%
2916                 \csname \curr@fontshape/\f@size/\CJK@plane\endcsname\fi}}}%
2917     \@gobble
2918 }%
2919 }{\@firstofone}%

```

This is the normal L^AT_EX definition.

```

2920 {\def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}}%

```

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

```

2921 \ifx\pickup@font\MT@orig@pickupfont \else
2922 \MT@warning@nl{%
2923     Command \string\pickup@font\space is not defined as expected.%
2924     \MessageBreak Patching it anyway. Some things may break%
2925 }{*package}
2926 .\MessageBreak Double-check whether micro-typography is indeed%
2927 \MessageBreak applied to the document.%
2928 \MessageBreak (Hint: Turn on `verbose' mode)%
2929 }{/package}
2930 }%
2931 \fi

```

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

```

2932 \g@addto@macro\pickup@font{\begingroup}%

```

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

```

2933 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
2934 \g@addto@macro\pickup@font{%
2935     \escapechar\m@ne
2936 }{*package}
2937 {debug} \global\MT@inannottrue
2938 {debug} \MT@gl@et\MT@pdf@annot\@empty
2939 {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.

```

2940 \MT@let@cn\MT@font\MT@subst\expandafter\string\font@name}%
2941 \ifx\MT@font\relax
2942 \let\MT@font\font@name
2943 \else
2944 \ifx\MT@font\font@name \else
2945 {debug} \MT@addto@annot{= substituted with \MT@font}%
2946 \MT@register@subst@font
2947 \fi
2948 \fi
2949 \MT@setupfont
2950 }{/package}
2951 {letterspace} \MT@tracking
2952 \endgroup
2953 }%
2954 }{*package}

```

<code>\MT@pickupfont</code>	Remember the patched command, because we may have to disable ourselves in certain situations.
<code>\MT@MT@pickupfont</code>	
<code>\MT@ltx@pickupfont</code>	<pre> 2955 \let\MT@pickupfont\pickup@font 2956 \def\MT@MT@pickupfont {\let\pickup@font\MT@pickupfont}% 2957 \def\MT@ltx@pickupfont{\let\pickup@font\MT@orig@pickupfont}% </pre>
<code>\do@subst@correction</code>	<p>Additionally, we hook into <code>\do@subst@correction</code>, 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.</p> <pre> 2958 \g@addto@macro\do@subst@correction 2959 { \edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}% 2960 \MT@glet@nc{MT@subst@expandafter\string\font@name}\MT@font}% </pre>
<code>\add@accent</code>	Inside <code>\add@accent</code> , we have to disable microtype's setup, since the grouping in the patched <code>\pickup@font</code> would break the accent if different fonts are used for the base character and the accent. Fortunately, L ^A T _E X takes care that the fonts used for the <code>\accent</code> are already set up, so that we cannot be overlooking them.
<code>\MT@orig@add@accent</code>	<pre> 2961 \let\MT@orig@add@accent\add@accent 2962 \def\add@accent#1#2{% 2963 \MT@ltx@pickupfont 2964 \MT@orig@add@accent{#1}{#2}% 2965 \MT@MT@pickupfont 2966 }% 2967 <i>/package</i> 2968 } 2969 <i>plain</i> \relax 2970 <i>*package</i> </pre> <p>Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.</p>
<code>\MT@check@font</code>	Check whether we've already seen the current font.
	<pre> 2971 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list} </pre>
<code>\MT@register@font</code>	Register the current font.
	<pre> 2972 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}} </pre>
<code>\MT@register@subst@font</code>	Register the substituted font (only if it isn't registered already). Additionally, we have to remove the substitute font from the list of fonts, so that we set it up again.
	<pre> 2973 \def\MT@register@subst@font{% 2974 \MT@exp@one@n\MT@in@clist\font@name\MT@font@list 2975 \ifMT@inlist@else 2976 \xdef\MT@font@list{\MT@font@list\font@name,}% 2977 \expandafter\MT@rem@from@clist\MT@font\MT@font@list 2978 \fi 2979 } </pre>

14.2.10 Context-sensitive setup

	Here are the variants for context-sensitive setup.
<code>\MT@active@features</code>	The activated features are stored in this command.
	<pre> 2980 \let\MT@active@features\empty </pre>
<code>\MT@check@font@cx</code>	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.

```

2981 \def\MT@check@font@cx{%
2982   \MT@if@true
2983   \MT@map@clist@c\MT@active@features{%
2984     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
2985     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
2986     \ifMT@inlist@
2987       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
2988     \else
2989       \MT@if@false
2990     \fi
2991   }%
2992   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2993 }

```

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

```

2994 \def\MT@register@subst@font@cx{%
2995   \MT@map@clist@c\MT@active@features{%
2996     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name
2997     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
2998     \ifMT@inlist@ \else
2999       \MT@exp@cs\MT@xadd
3000       {MT@##1\csname MT@##1@context\endcsname font@list}%
3001       {\font@name,%}
3002       \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter\MT@font
3003       \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
3004     \fi
3005   }%
3006 }

```

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

```

3007 \def\MT@register@font@cx{%
3008   \MT@map@clist@c\MT@active@features{%
3009     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
3010     \MT@exp@cs\MT@xadd
3011     {MT@##1\csname MT@##1@context\endcsname font@list}%
3012     {\MT@font,%}
3013     \def\@tempa{##1}%
3014     \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
3015   \fi
3016 }%
3017 }

```

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

```

3018 \def\MT@maybe@rem@from@list#1{%
3019   \MT@if@fstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
3020     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
3021     \MT@font \csname MT@\@tempa @#1font@list\endcsname
3022   }%
3023 }

```

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

Inside the preamble, this command shouldn't actually do anything but remember itself for later.

```

3024 \def\microtypecontext{\MT@begin@catcodes\MT@microtypecontext}
3025 \def\MT@microtypecontext#1{\MT@end@catcodes\MT@addto@setup{\microtypecontext{#1}}}
3026 \MT@addto@setup{%
3027   \DeclareRobustCommand\microtypecontext{%
3028     \MT@begin@catcodes
3029     \MT@microtypecontext
3030   }%
3031   \def\MT@microtypecontext#1{%
3032     \MT@end@catcodes

```

```

3033 \MT@setup@contexts
3034 \let\MT@reset@context\relax

```

We need to ensure that math fonts are set up anew.

```

3035 \MT@glet\glb@currsizel@empty
3036 \setkeys{MTC}{#1}%
3037 \selectfont
3038 \MT@reset@context
3039 }%
3040 }

```

\textmicrotypecontext This is just a wrapper around \microtypecontext.

```

\MT@textmicrotypecontext 3041 \DeclareRobustCommand\textmicrotypecontext{\MT@begin@catcodes\MT@textmicrotypecontext}
\MT@textmicrotypecontext 3042 \def\MT@textmicrotypecontext#1{\MT@end@catcodes\MT@textmicrotypecontext{#1}}
3043 \def\MT@textmicrotypecontext#1#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.

```

3044 \def\MT@reset@context@{%
3045 \MT@vinfo{<<< Resetting contexts\on@line
3046 <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
3047 <debug> / \MT@tr@context/\MT@kn@context/\MT@sp@context
3048 }%
3049 \selectfont
3050 }

```

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

```

3051 \def\MT@setup@contexts{%
3052 \MT@map@clist@c\MT@active@features
3053 {\MT@glet@c\MT@#1@font@list}\MT@font@list}%
3054 \MT@glet\MT@check@font\MT@check@font@cx
3055 \MT@glet\MT@register@font\MT@register@font@cx
3056 \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
3057 \MT@glet\MT@setup@contexts\relax
3058 }

```

Define context keys.

```

3059 \MT@map@clist@c\MT@features@long{%
3060 \define@key{MTC}{#1}[]{}%
3061 \edef\@tempb{\@nameuse{MT@rbba@#1}}%
3062 \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
3063 \ifMT@inlist@

```

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

```

3064 \MT@ifempty{##1}{\def\MT@val{0}}{\def\MT@val{##1}}%
3065 \MT@exp@cs\ifx\MT@tempb @context\MT@val
3066 <debug> \MT@dinfo{1}{>>> no change of #1 context: \MT@val}%
3067 \else
3068 \MT@vinfo{>>> Changing #1 context to \MT@val'\MessageBreak\on@line
3069 <debug> \space{previous: \@nameuse{MT@tempb @context}'}%
3070 }%
3071 \def\MT@reset@context{\aftergroup\MT@reset@context@}%

```

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

```

3072 \MT@glet@nn{\MT@reset@\@tempb @codes}\MT@reset@\@tempb @codes}%

```

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

```

3073 \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
3074 \MT@val \csname MT@\@tempb @doc@contexts\endcsname
3075 \ifMT@inlist@ \else
3076 \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{\MT@val}%
3077 <debug> \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%

```



```

3078         \fi
3079         \MT@edef@n{MT@ \@tempb @context}{\MT@val}%
3080         \fi
3081     \fi
3082 }%
3083 }

```

We also allow the activate shortcut.

```

3084 \define@key{MTC}{activate}[]{}%
3085 \setkeys{MTC}{protrusion={#1}}%
3086 \setkeys{MTC}{expansion={#1}}%
3087 }

```

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

`\MT@ex@context` 3088 `\MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%`

`\MT@tr@context` 3089 `\MT@def@n{MT@#1@context}{@}%`

`\MT@sp@context` 3090 `\MT@def@n{MT@#1@doc@contexts}{{@}}%`

`\MT@kn@context` 3091 }

`\MT@kn@context` 3092 `\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.

```

3093 \def\DeclareMicrotypeSet{%
3094     \MT@begin@catcodes
3095     \ifstar
3096         \MT@DeclareSetAndUseIt
3097         \MT@DeclareSet
3098     }

```

`\MT@DeclareSet`

```

3099 \newcommand\MT@DeclareSet[3][]{%
3100     \MT@ifempty{#1}{%
3101         \MT@map@clist@c\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%
3102     }{%
3103         \MT@map@clist@n{#1}{%
3104             \MT@ifempty{##1}\relax{%
3105                 \MT@is@feature{##1}{set declaration `#2'}{%
3106                     \MT@exp@one@n\MT@declare@sets
3107                     {\csname MT@rbba@##1\endcsname}{#2}{#3}%
3108                 }%
3109             }%
3110         }}%
3111     }%
3112     \MT@end@catcodes
3113 }

```

`\MT@DeclareSetAndUseIt`

```

3114 \newcommand\MT@DeclareSetAndUseIt[3][]{%
3115     \MT@DeclareSet[#1]{#2}{#3}%
3116     \UseMicrotypeSet[#1]{#2}%
3117 }

```

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

3118 `\let\MT@curr@set@name\@empty`

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

```

3119 \def\MT@declare@sets#1#2#3{%
3120   \def\MT@curr@set@name{#2}%
3121   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
3122     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
3123     \MT@map@clist@n{font,encoding,family,series,shape,size}{%
3124       \MT@glet@nc{MT@#1list@##1@\MT@curr@set@name}\@undefined
3125     }%
3126   }%
3127   \MT@glet@nc{MT@#1@set@@\MT@curr@set@name}\@empty
3128   (debug)\MT@edinfo{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
3129   \setkeys{MT@#1@set}{#3}%
3130 }
```

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

```

3131 \def\MT@define@set@key@#1#2{%
3132   \define@key{MT@#2@set}{#1}[]{}%
3133   \MT@glet@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
3134   \MT@map@clist@n{##1}{%
3135     \KV@esp@def\MT@val{###1}%
3136     \MT@get@highlevel{#1}%

```

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

```

3137     \MT@exp@two@n@g@addto@macro
3138     {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
3139     {\MT@val},}%
3140   }%

```

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

```

3141   \expandafter\g@addto@macro\expandafter\MT@font@sets
3142   \csname MT@#2list@#1@\MT@curr@set@name\endcsname
3143   (debug)\MT@edinfo@n1{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
3144   }%
3145 }
```

`\MT@get@highlevel` Saying, for instance, ‘family=rm*’ or ‘shape=bf*’ will expand to `\rmdefault` resp. `\bfdefault`.

```

3146 \def\MT@get@highlevel#1{%
3147   \expandafter\MT@test@ast\MT@val*\@nil\relax}%

```

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

```

3148   \MT@ifempty\@tempa{\def\@tempa{#1}}\relax

```

Test whether the command is actually defined.

```

3149   \MT@ifdefined@n@TF{\@tempa default}%
3150   {\edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}}%
3151   {\MT@warning{~\@backslashchar\@tempa default' is not a defined command.\MessageBreak
3152     Ignoring `#1 = {\@tempa*}' in font set\MessageBreak`~\MT@curr@set@name'}%
3153     \let\MT@val\@empty}%

```

In contrast to earlier versions, these values will not be expanded immediately, but at the end of the preamble.

```

3154   }%
3155 }
```

`\MT@test@ast` It the last character is an asterisk, execute the second argument, otherwise the first one.

```

3156 \def\MT@test@ast#1*#2\@nil{%
3157   \def\@tempa{#1}%
3158   \MT@ifempty{#2}%
3159 }
```

`\MT@font@sets` Fully expand the font specification and fix catcodes for all font sets. Also remove

`\MT@fix@font@set`

fontspec's counters.

```

3160 \let\MT@font@sets\empty
3161 \def\MT@fix@font@set#1{%
3162   \MT@ifdefined@c@T{#1}{%
3163     \xdef#1{#1}%
3164     \ifMT@fontspec
3165       \xdef#1{\expandafter\MT@scrubfeatures#1()\relax}%
3166     \fi
3167     \global\@onelevel@sanitize#1%
3168   }%
3169 }
```

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

```

3170 \def\MT@define@set@key@size#1{%
3171   \define@key{MT@#1@set}{size}[]{%
3172     \MT@map@cliston{##1}{%
3173       \def\MT@val{####1}%
3174       \expandafter\MT@get@range\MT@val--\@nil
3175       \ifx\MT@val\relax \else
3176         \MT@exp@cs\MT@xadd
3177         {MT@#1list@size@MT@curr@set@name}%
3178         {{\MT@lower}{\MT@upper}\relax}}%
3179     \fi
3180   }%
3181   <debug>\MT@infoon{1}{-- size: \@nameuse{MT@#1list@size@MT@curr@set@name}}%
3182 }%
3183 }
```

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 does this for the OpenType version of Adobe's Minion. (Available from CTAN at [pkg/minionpro](#)))

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

```

\MT@lower 3184 \def\MT@get@range#1-#2-#3\@nil{%
3185   \MT@ifempty{#1}{%
3186     \MT@ifempty{#2}{%
3187       \let\MT@val\relax
3188     }%
3189     \def\MT@lower{0}%
3190     \def\MT@val{#2}%
3191     \MT@get@size
3192     \edef\MT@upper{\MT@val}%
3193   }%
3194 }%
3195   \def\MT@val{#1}%
3196   \MT@get@size
3197   \ifx\MT@val\relax \else
3198     \edef\MT@lower{\MT@val}%
3199     \MT@ifempty{#2}{%
3200       \MT@ifempty{#3}%
3201       {\def\MT@upper{-1}}%

```

2048 pt is T_EX's maximum font size.

```

3202   {\def\MT@upper{2048}}%
3203 }%
3204   \def\MT@val{#2}%
3205   \MT@get@size
3206   \ifx\MT@val\relax \else
3207     \MT@ifdim\MT@lower>\MT@val{%
3208       \MT@error{%
3209         Invalid size range (\MT@lower\space > \MT@val) in font set

```

```

3210         \MT@curr@set@name'.\MessageBreak Swapping sizes}}}%
3211         \edef\MT@upper{\MT@lower}%
3212         \edef\MT@lower{\MT@val}%
3213     }{%
3214         \edef\MT@upper{\MT@val}%
3215     }%
3216     \MT@ifdim\MT@lower=\MT@upper
3217     {\def\MT@upper{-1}}%
3218     \relax
3219 \fi
3220 }%
3221 \fi
3222 }%
3223 }

```

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

```

3224 \def\MT@get@size{%
    A single star would mean \sizedefault, which doesn't exist, so we define it to be
    \normalsize.
3225     \if*\MT@val\relax
3226         \def\@tempa{\normalsize}%
3227     \else
3228         \MT@let@cn\@tempa{\MT@val}%
3229     \fi
3230     \ifx\@tempa\relax\else
3231         \MT@get@size@
3232     \fi

```

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

```

3233     \MT@ifdimen\MT@val{%
3234         \@defaultunits\@tempdima\MT@val pt\relax\@nnil
3235         \edef\MT@val{\strip@pt\@tempdima}%
3236     }{%
3237         \MT@warning{Could not parse font size `'\MT@val'\MessageBreak
3238                     in font set `'\MT@curr@set@name'}%
3239         \let\MT@val\relax
3240     }%
3241 }

```

`\MT@get@size@` 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` instead of `\@setfontsize` because some classes might define the size selection commands by simply using `\fontsize` (e.g., the `a0poster` class).

```

3242 \def\MT@get@size@@{%
3243     \begingroup
3244     \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
3245     \@tempa\@nil
3246 }

```

The `svjour3` class defines the size commands using conditionals; using e-TeX primitives, we close any leftovers here.

```

3247 ^^X\@ifclassloaded{svjour3}{%
3248 ^^X     \def\MT@get@size@{%
3249 ^^X         \@tempcnta=\currentiflevel
3250 ^^X         \MT@get@size@@
3251 ^^X         \MT@loop
3252 ^^X             \ifnum\numexpr\currentiflevel-1>\@tempcnta
3253 ^^X                 \csname fi\endcsname
3254 ^^X             \MT@repeat
3255 ^^X         }%
3256 ^^X}%

```

```

3257 \let\MT@get@size@\MT@get@size@@
3258 ~X}

```

\MT@define@set@key@font

```

3259 \def\MT@define@set@key@font#1{%
3260 \define@key{MT@#1@set}{font}[]{%
3261 \MT@get@nc{MT@#1list@font@\MT@curr@set@name}\@empty
3262 \MT@map@clist@n{##1}{%
3263 \def\MT@val{###1}%
3264 \MT@ifstreq\MT@val*{\def\MT@val{*/**/*/*}}\relax
3265 \expandafter\MT@get@font\MT@val////\@nil
3266 \MT@exp@two@n@g@addto@macro
3267 {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
3268 {\MT@val},}%
3269 }%
3270 \expandafter\g@addto@macro\expandafter\MT@font@sets
3271 \csname MT@#1list@font@\MT@curr@set@name\endcsname
3272 <debug>\MT@edinfo@n{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
3273 }%
3274 }

```

\MT@get@font Translate any asterisks.

```

3275 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
3276 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
3277 \ifx\MT@val\relax\def\MT@val{0}\fi
3278 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
3279 \let\MT@val\@tempb
3280 }

```

\MT@get@font@ Helper macro, also used by \MT@get@font@and@size.

```

3281 \def\MT@get@font@#1#2#3#4#5#6{%
3282 \let\@tempb\@empty
3283 \def\MT@temp{#1/#2/#3/#4/#5}%
3284 \MT@get@axis{encoding}{#1}%
3285 \MT@get@axis{family}{#2}%
3286 \MT@get@axis{series}{#3}%
3287 \MT@get@axis{shape}{#4}%
3288 \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
3289 \MT@ifempty{#5}{%
3290 \MT@warn@axis@empty{size}{\string\normalsize}%
3291 \def\MT@val{*}%
3292 }{%
3293 \def\MT@val{#5}%
3294 }%
3295 \MT@get@size
3296 }

```

\MT@get@axis

```

3297 \def\MT@get@axis#1#2{%
3298 \def\MT@val{#2}%
3299 \MT@get@highlevel{#1}%
3300 \MT@ifempty\MT@val{%
3301 \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
3302 \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
3303 }\relax
3304 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
3305 }

```

\MT@warn@axis@empty

```

3306 \def\MT@warn@axis@empty#1#2{%
3307 \MT@warning{#1 axis is empty in font specification\MessageBreak
3308 ~\MT@temp'. Using `#2' instead}%
3309 }

```

We can finally assemble all pieces to define \DeclareMicrotypeSet's keys. They are

also used for \DisableLigatures.

```

3310 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
3311 \MT@define@set@key@{encoding}{#1}%
3312 \MT@define@set@key@{family}{#1}%
3313 \MT@define@set@key@{series}{#1}%
3314 \MT@define@set@key@{shape}{#1}%
3315 \MT@define@set@key@size{#1}%
3316 \MT@define@set@key@font{#1}%
3317 }

```

\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.

```

3318 \def\UseMicrotypeSet{%
3319 \MT@begin@catcodes
3320 \MT@UseMicrotypeSet
3321 }

```

\MT@UseMicrotypeSet

```

3322 \newcommand*\MT@UseMicrotypeSet[2][ ]{%
3323 \MT@ifempty{#1}{%
3324 \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
3325 }{%
3326 \MT@map@clist@n{#1}{%
3327 \MT@ifempty{##1}\relax{%
3328 \MT@is@feature{##1}{activation of set `#2'}{%
3329 \MT@exp@one@n\MT@use@set
3330 {\csname MT@rbba@##1\endcsname}{#2}%
3331 }%
3332 }%
3333 }}%
3334 }%
3335 \MT@end@catcodes
3336 }

```

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

```

\MT@ex@setname 3337 \def\MT@use@set#1#2{%
\MT@tr@setname 3338 \MT@ifdefined@n@TF{MT@#1@set@#2}{%
\MT@xdef@n{MT@#1@setname}{#2}%
\MT@sp@setname 3339 }{%
\MT@kn@setname 3340 }%
\MT@use@set 3341 \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
3342 \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
3343 }%
3344 \MT@error{%
3345 The \@nameuse{MT@abbr@#1} set `#2' is undeclared.\MessageBreak
3346 Using set \@nameuse{MT@#1@setname}' instead}{}%
3347 }%
3348 }

```

\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.

```

3349 \def\DeclareMicrotypeSetDefault{%
3350 \MT@begin@catcodes
3351 \MT@DeclareMicrotypeSetDefault
3352 }

```

\MT@DeclareMicrotypeSetDefault

```

3353 \newcommand*\MT@DeclareMicrotypeSetDefault[2][ ]{%
3354 \MT@ifempty{#1}{%
3355 \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
3356 }{%
3357 \MT@map@clist@n{#1}{%
3358 \MT@ifempty{##1}\relax{%
3359 \MT@is@feature{##1}{declaration of default set `#2'}{%
3360 \MT@exp@one@n\MT@set@default@set

```

```

3361         {\csname MT@rbba@##1\endcsname}{#2}%
3362     }%
3363 }%
3364 }}%
3365 }%
3366 \MT@end@catcodes
3367 }

\MT@default@pr@set
\MT@default@ex@set 3368 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 3369 \MT@ifdefined@n@TF{MT@#1@set@#2}{%
3370 debug\MT@edinfo{1}{declaring default \@nameuse{MT@abbr@#1} set `#2'}%
\MT@default@sp@set 3371 \MT@xdef@n{MT@default@#1@set}{#2}%
\MT@default@kn@set 3372 }{%
3373 \MT@error{%
3374     The \@nameuse{MT@abbr@#1} set `#2' is not declared.\MessageBreak
3375     Cannot make it the default set. Using set\MessageBreak `all' instead}{}%
3376 \MT@xdef@n{MT@default@#1@set}{all}%
3377 }%
3378 }

```

14.3.2 Variants and aliases

`\DeclareMicrotypeVariants` Specify suffixes for variants (see `fontname/variants.map`). The starred version appends to the list.

```

\MT@variants
3379 \let\MT@variants\@empty
3380 \def\DeclareMicrotypeVariants{%
3381     \MT@begin@catcodes
3382     \ifstar
3383         \MT@DeclareVariants
3384     {\let\MT@variants\@empty\MT@DeclareVariants}%
3385 }

```

`\MT@DeclareVariants`

```

3386 \def\MT@DeclareVariants#1{%
3387     \MT@map@clist@n{#1}%
3388     \def\@tempa{##1}%
3389     \@onelevel@sanitize\@tempa
3390     \xdef\MT@variants{\MT@variants{\@tempa}}%
3391 }%
3392 \MT@end@catcodes
3393 }

```

`\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.

```

3394 \def\DeclareMicrotypeAlias{%
3395     \MT@begin@catcodes
3396     \MT@DeclareMicrotypeAlias
3397 }

```

`\MT@DeclareMicrotypeAlias`

```

3398 \newcommand*\MT@DeclareMicrotypeAlias[2]{%
3399     \def\@tempb{#2}%
3400     \@onelevel@sanitize\@tempb
3401     \MT@ifdefined@n@T{MT@#1@alias}{%
3402         \MT@warning{Alias font family `\'@tempb' will override
3403             alias `\'@nameuse{MT@#1@alias}'\MessageBreak
3404             for font family `#1'}%
3405     \MT@xdef@n{MT@#1@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.

```

3406 \MT@ifdefined@c@T\MT@family{%
3407 (debug)\MT@edinfo{1}{Activating alias font '\@tempb' for '\MT@family'}%
3408 \MT@glet\MT@familyalias\@tempb
3409 }%
3410 \MT@end@catcodes
3411 }

```

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

```

3412 \def\LoadMicrotypeFile#1{%
3413 \edef\@tempa{\zap@space#1 \@empty}%
3414 \@onelevel@sanitize\@tempa
3415 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
3416 \ifMT@inlist@
3417 \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
3418 \else
3419 \MT@xadd\MT@file@list{\@tempa,}%
3420 \MT@begin@catcodes
3421 \InputIfFileExists{mt-\@tempa.cfg}{%
3422 \edef\MT@curr@file{mt-\@tempa.cfg}%
3423 \MT@vinfo{... Loading configuration file \MT@curr@file}%
3424 }{%
3425 \MT@warning{Configuration file mt-\@tempa.cfg\MessageBreak
3426 does not exist}%
3427 }%
3428 \MT@end@catcodes
3429 \fi
3430 }
3431 (/package)
3432 (/package|letterspace)

```

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@nl@setname` The optional argument may be used to disable selected ligatures only.

```

\MT@nl@ligatures 3433 (*pdfTeX-def|LaTeX-def)
3434 (pdfTeX-def)\MT@requires@pdfTeX5{
3435 \def\DisableLigatures{%
3436 \MT@begin@catcodes
3437 \MT@DisableLigatures
3438 }
3439 \newcommand*\MT@DisableLigatures[2][]{%
3440 \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
3441 \xdef\MT@active@features{\MT@active@features,nl}%
3442 \global\MT@noligaturestrue
3443 \MT@declare@sets{nl}{no ligatures}{#2}%
3444 \gdef\MT@nl@setname{no ligatures}%
3445 \MT@end@catcodes
3446 }
3447 (pdfTeX-def){
3448 (/pdfTeX-def|LaTeX-def)

```

If pdf_TE_X is too old, we throw an error.

```

3449 (*pdfTeX-def|xetex-def)
3450 \renewcommand*\DisableLigatures[2][]{%
3451 \MT@error{Disabling ligatures of a font is only possible\MessageBreak
3452 with pdfTeX version 1.30 or newer.\MessageBreak
3453 Ignoring \string\DisableLigatures}%
3454 (pdfTeX-def) Upgrade
3455 (xetex-def) Use
3456 pdfTeX.}%
3457 }

```



```

3458 <pdf $\text{tex}$ -def>
3459 </pdf $\text{tex}$ -def| $\text{xetex}$ -def>

```

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.

```

3460 <*package>
3461 \def\DeclareMicrotypeBabelHook#1#2{%
3462   \MT@map@clist@n{#1}%
3463   \KV@sp@def\@tempa{##1}%
3464   \MT@gdef@n{MT@babel@{\@tempa}{#2}%
3465   }%
3466 }
3467 </package>

```

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).

```

3468 <*pdf $\text{tex}$ -def| $\text{xetex}$ -def| $\text{luatex}$ -def>
3469 \def\SetProtrusion{%
3470   \MT@begin@catcodes
3471   \MT@SetProtrusion
3472 }

```

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

```

\MT@pr@c@name 3473 \newcommand*\MT@SetProtrusion[3][]{%

```

```

\MT@extra@context 3474 \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.

```

3475 \MT@set@named@keys{MT@pr@c}{#1}%
3476 <debug>\MT@dinfor{1}{creating protrusion list ~\MT@pr@c@name'%
3477 \def\MT@permutelist{pr@c}%
3478 \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>`, ...

```

3479 \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.

```

3480 \MT@gdef@n{MT@pr@c@MT@pr@c@name}{#3}%
3481 \MT@end@catcodes
3482 }
3483 </pdf $\text{tex}$ -def| $\text{xetex}$ -def| $\text{luatex}$ -def>

```

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

```

3484 <*pdf $\text{tex}$ -def| $\text{luatex}$ -def>
3485 \def\SetExpansion{%
3486   \MT@begin@catcodes
3487   \MT@SetExpansion
3488 }

```

```

\MT@SetExpansion
  \MT@ex@c@name 3489 \newcommand*\MT@SetExpansion[3] [] {%
\MT@extra@context 3490 \let\MT@extra@context\@empty
  3491 \MT@set@named@keys{MT@ex@c}{#1}%
\MT@permutelist 3492 \MT@ifdefined@n@T{MT@ex@c@MT@ex@c@name @factor}{%
  3493 \ifnum\csname MT@ex@c@MT@ex@c@name @factor\endcsname > \@m
  3494 \MT@warning@n1{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
  3495 too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
  3496 maximum of 1000}%
  3497 \MT@gl@et@nc{MT@ex@c@MT@ex@c@name @factor}\@m
  3498 \fi
  3499 }%
  3500 <debug>\MT@dinfor{1}{creating expansion list `~\MT@ex@c@name'}%
  3501 \def\MT@permutelist{ex@c}%
  3502 \setkeys{MT@c@fg}{#2}%
  3503 \MT@permute
  3504 \MT@gdef@n{MT@ex@c@MT@ex@c@name}{#3}%
  3505 \MT@end@catcodes
  3506 }

\SetTracking
  3507 \def\SetTracking{%
  3508 \MT@begin@catcodes
  3509 \MT@SetTracking
  3510 }

\MT@SetTracking Third argument may be empty.
  3511 \newcommand*\MT@SetTracking[3] [] {%
  3512 \let\MT@extra@context\@empty
  3513 \MT@set@named@keys{MT@tr@c}{#1}%
  3514 <debug>\MT@dinfor{1}{creating tracking list `~\MT@tr@c@name'}%
  3515 \def\MT@permutelist{tr@c}%
  3516 \setkeys{MT@c@fg}{#2}%
  3517 \MT@permute
  3518 \KV@sp@def\@tempa{#3}%
  3519 \MT@ifempty\@tempa\relax{%
  3520 \MT@ifint\@tempa
  3521 {\MT@xdef@n{MT@tr@c@MT@tr@c@name}{\@tempa}}%
  3522 {\MT@warning{Value `~\@tempa' is not a number in\MessageBreak
  3523 tracking set `~\MT@curr@set@name'}}}%
  3524 \MT@end@catcodes
  3525 }
  3526 </pdfTeX-def|luatex-def>

\SetExtraSpacing
  3527 <*pdfTeX-def>
  3528 \def\SetExtraSpacing{%
  3529 \MT@begin@catcodes
  3530 \MT@SetExtraSpacing
  3531 }

\MT@SetExtraSpacing
  \MT@sp@c@name 3532 \newcommand*\MT@SetExtraSpacing[3] [] {%
\MT@extra@context 3533 \let\MT@extra@context\@empty
  3534 \MT@set@named@keys{MT@sp@c}{#1}%
\MT@permutelist 3535 <debug>\MT@dinfor{1}{creating spacing list `~\MT@sp@c@name'}%
  3536 \def\MT@permutelist{sp@c}%
  3537 \setkeys{MT@c@fg}{#2}%
  3538 \MT@permute
  3539 \MT@gdef@n{MT@sp@c@MT@sp@c@name}{#3}%
  3540 \MT@end@catcodes
  3541 }

\SetExtraKerning

```

```

3542 \def\SetExtraKerning{%
3543   \MT@begin@catcodes
3544   \MT@SetExtraKerning
3545 }

```

\MT@SetExtraKerning

```

\MT@kn@c@name 3546 \newcommand*\MT@SetExtraKerning[3] [] {%
3547   \let\MT@extra@context\empty
\MT@extra@context 3548   \MT@set@named@keys{\MT@kn@c}{#1}%
\MT@permute@list 3549   (debug)\MT@dinfor{1}{creating kerning list '\MT@kn@c@name'}%
3550   \def\MT@permute@list{\MT@kn@c}%
3551   \setkeys{\MT@cfg}{#2}%
3552   \MT@permute
3553   \MT@gdefon{\MT@kn@c@{\MT@kn@c@name}}{#3}%
3554   \MT@end@catcodes
3555 }
3556 (/pdfTeX-def)

```

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

```

\MT@options 3557 (*package)
3558 \def\MT@set@named@keys#1#2{%
3559   \def\x##1name=##2,##3\@nil{%
3560     \setkeys{#1}{name=##2}%
3561     \gdef\MT@options{##1##3}%
3562     \MT@rem@from@clist{name=}\MT@options
3563   }%
3564   \x#2,name=,\@nil
3565   \expandtwoargs\setkeys{#1}\MT@options
3566 }

```

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

```

3567 \def\MT@define@code@key#1#2{%
3568   \define@key{\MT@#2}{#1} [] {%
3569     \@tempcnta=\@ne
3570     \MT@map@cliston{##1}%
3571     \KV@@sp@def\MT@val{###1}%

```

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

```

3572     \MT@get@highlevel{#1}%
3573     \MT@edefon{\MT@temp#1\the\@tempcnta}{\MT@val}%
3574     \advance\@tempcnta \@ne
3575   }%
3576 }%
3577 }

```

\MT@define@code@key@family Remove fontspec’s internal feature counter.

```

3578 \def\MT@define@code@key@family#1{%
3579   \define@key{\MT@#1}{family} [] {%
3580     \@tempcnta=\@ne
3581     \MT@map@cliston{##1}%
3582     \KV@@sp@def\MT@val{###1}%
3583     \MT@get@highlevel{family}%
3584     \ifMT@fontspec
3585       \edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x
3586     \fi
3587     \MT@edefon{\MT@tempfamily\the\@tempcnta}{\MT@val}%
3588     \advance\@tempcnta \@ne
3589   }%
3590 }%
3591 }

```

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

```

3592 \def\MT@define@code@key@size#1{%
3593   \define@key{MT@#1}{size}[]{%
3594     \MT@map@clist@n{##1}{%
3595       \KV@esp@def\MT@val{###1}%
3596       \expandafter\MT@get@range\MT@val--\@nil
3597       \ifx\MT@val\relax \else
3598         \MT@exp@cs\MT@xadd\MT@tempsize{%
3599           {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
3600       \fi
3601     }%
3602   }%
3603 }

```

\MT@define@code@key@font

```

3604 \def\MT@define@code@key@font#1{%
3605   \define@key{MT@#1}{font}[]{%
3606     \MT@map@clist@n{##1}{%
3607       \KV@esp@def\MT@val{###1}%
3608       \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/}}\relax
3609       \expandafter\MT@get@font@and@size\MT@val////\@nil
3610       \ifMT@fontspec
3611         \edef\@tempb{\expandafter\MT@scrubfeatures\@tempb()\relax}%
3612       \fi
3613       \MT@xdef@n{\MT@MT@permutelist @\@tempb\MT@extra@context}%
3614       {\csname MT@\MT@permutelist @name\endcsname}%
3615       (debug) \MT@info@n{1}{initialising: use list for font \@tempb=\MT@val}
3616       (debug) \ifx\MT@extra@context\empty\else\MessageBreak
3617       (debug) (context: \MT@extra@context)\fi}%
3618       \MT@exp@cs\MT@xaddb
3619       {\MT@MT@permutelist @\@tempb\MT@extra@context @size}%
3620       {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
3621     }%
3622   }%
3623 }

```

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

```

3624 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
3625   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
3626 }

3627 \MT@define@code@key{encoding}{cfg}
3628 \MT@define@code@key{family} {cfg}
3629 \MT@define@code@key{series} {cfg}
3630 \MT@define@code@key{shape} {cfg}
3631 \MT@define@code@key@size {cfg}
3632 \MT@define@code@key@font {cfg}

```

\MT@define@opt@key

```

3633 \def\MT@define@opt@key#1#2{%
3634   \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
3635     \MT@xdef@n{MT@#1@c@MT@curr@set@name @#2}{##1}}}%
3636 }

```

\MT@listname@count The options in the optional first argument.

```

3637 \newcount\MT@listname@count
3638 \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 – also check whether the name already exists (in case more than one unnamed list is loaded in the same line, for example \AtBeginDocument).

```

3639   \define@key{MT@#1@c}{name}[]{%
3640     \MT@ifempty{##1}{%
3641       \MT@ifdefined@nTF{MT@#1@c@MT@curr@file/\the\inputlineno}{%
3642         \global\advance\MT@listname@count\@ne
3643         \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno

```

```

3644                                     (\number\MT@listname@count))}%
3645     }{%
3646       \MT@edef\MT@#1@c@name{\MT@curr@file/\the\inputlineno}%
3647     }%
3648   }{%
3649     \MT@edef\MT@#1@c@name{##1}%
3650     \MT@ifdefined\MT@#1@c@\csname MT@#1@c@name\endcsname{%
3651       \MT@warning{Redefining \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}}}%
3652     }%
3653   }%
3654   \MT@let\MT@curr@set@name{MT@#1@c@name}%
3655 }%
3656 \MT@define@opt@key{#1}{load}%
3657 \MT@define@opt@key{#1}{factor}%
3658 \MT@define@opt@key{#1}{preset}%
3659 \MT@define@opt@key{#1}{inputenc}%

```

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

```

3660 \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
3661 }
3662 /package

```

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. It also works with LuaTeX 0.30 or newer.

```

3663 (*pdfTeX-def|luatex-def)
3664 (pdfTeX-def)\MT@requires@pdfTeX7{
3665   \define@key{MT@ex@c}{context}[]{%
3666     \MT@ifempty{#1}\relax{%
3667       \MT@gl@t\MT@copy@font\MT@copy@font@
3668       \def\MT@extra@context{#1}%
3669     }%
3670   }
3671   \MT@addto@setup{%
3672     \define@key{MT@ex@c}{context}[]{%
3673       \ifx\MT@copy@font\MT@copy@font@
3674         \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}}%
3675     }else
3676       \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
3677         Ignoring `context' key\on@line}%
3678       {Either move the settings inside the preamble,\MessageBreak
3679         or load the package with the `copyfonts' option.}%
3680     \fi
3681   }%
3682 }

```

Protrusion contexts *might* 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.

```

3683 \define@key{MT@pr@c}{context}[]{%
3684   \MT@ifempty{#1}\relax{%
3685     \MT@gl@t\MT@copy@font\MT@copy@font@
3686     \def\MT@extra@context{#1}%
3687   }%
3688 }
3689 \MT@addto@setup{%
3690   \define@key{MT@pr@c}{context}[]{%
3691     \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}}%
3692   \ifx\MT@copy@font\MT@copy@font@
3693     \MT@warning{n!If protrusion contexts don't work as expected,

```

```

3694 \MessageBreak load the package with the `copyfonts' option}%
3695 \fi
3696 }%
3697 }
3698 </pdfTeX-def|luatex-def>
3699 <*pdfTeX-def>
3700 {}{
3701 \define@key{MT@ex@c}{context}[]{%
3702 \MT@error{Expansion contexts only work with pdfTeX 1.40.4\MessageBreak
3703 or later. Ignoring `context' key\on@line}%
3704 {Upgrade pdfTeX.}%
3705 }
3706 </pdfTeX-def>
3707 <*pdfTeX-def|xetex-def>
3708 \define@key{MT@pr@c}{context}[]{%
3709 \MT@error{Protrusion contexts only work with pdfTeX
3710 <pdfTeX-def> 1.40.4\MessageBreak or later.
3711 <xetex-def> \MessageBreak or luatex.
3712 Ignoring `context' key\on@line}%
3713 <pdfTeX-def> {Upgrade pdfTeX.}%
3714 <xetex-def> {Use pdfTeX or luatex.}%
3715 }
3716 </pdfTeX-def|xetex-def>
3717 <pdfTeX-def>}
```

\MT@warn@nodim

```

3718 <*package>
3719 \def\MT@warn@nodim#1{%
3720 \MT@warning{`@tempa' is not a dimension.\MessageBreak
3721 Ignoring it and setting values relative to\MessageBreak #1}%
3722 }
3723 </package>
```

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

```

3724 <*pdfTeX-def|xetex-def|luatex-def>
3725 \define@key{MT@pr@c}{unit}[character]{%
3726 \MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
3727 \def\@tempa{#1}%
3728 \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.

```

3729 \MT@ifdimen\@tempa
3730 {\MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
3731 {\MT@warn@nodim{character widths}}%
3732 }%
3733 }
3734 </pdfTeX-def|xetex-def|luatex-def>
```

Tracking may only be relative to a dimension.

```

3735 <*pdfTeX-def|luatex-def>
3736 \define@key{MT@tr@c}{unit}[1em]{%
3737 \MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
3738 \def\@tempa{#1}%
3739 \MT@ifdimen\@tempa
3740 {\MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
3741 {\MT@warn@nodim{1em}%
3742 \MT@gdef@n{MT@tr@c@MT@curr@set@name @unit}{1em}}%
3743 }
3744 </pdfTeX-def|luatex-def>
```

Spacing and kerning codes may additionally be relative to space dimensions.

```

3745 <*pdfTeX-def>
3746 \MT@map@clist@n{sp,kn}{%
3747 \define@key{MT@#1@c}{unit}[space]{%
```

```

3748 \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
3749 \def\@tempa{##1}%
3750 \MT@ifstreq\@tempa{character}\relax{%
3751 \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
3752 \MT@ifstreq\@tempa{space}\relax{%
3753 \MT@ifdimen\@tempa
3754 { \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
3755 { \MT@warn@nodim{width of space}}%
3756 }%
3757 }%
3758 }%
3759 }
3760 \pdfTeX-def

```

The first argument to `\SetExpansion` accepts some more options.

```

3761 \pdfTeX-def|luatex-def
3762 \MT@map@clist@n{stretch,shrink,step}%
3763 \define@key{MT@ex@c}{#1}[]{}%
3764 \MT@ifempty{##1}\relax{}%
3765 \MT@ifint{##1}%

```

A space terminates the number.

```

3766 \MT@gdef@n{MT@ex@c@MT@curr@set@name @#1}{##1}%
3767 }{}%
3768 \MT@warning{%
3769 Value `##1' for option `#1' is not a number.\MessageBreak
3770 Ignoring it}%
3771 }{}%
3772 }{}%
3773 }{}%
3774 }
3775 \define@key{MT@ex@c}{auto}[true]{}%
3776 \def\@tempa{#1}%
3777 \csname if\@tempa\endcsname

```

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

```

3778 \pdfTeX-def \MT@requires@pdfTeX4%
3779 \luatex-def \MT@requires@luatex3\relax
3780 { \MT@gdef@n{MT@ex@c@MT@curr@set@name @auto}{autoexpand}}%
3781 \pdfTeX-def { \MT@warning{pdfTeX too old for automatic font expansion}}%
3782 \else
3783 \pdfTeX-def \MT@requires@pdfTeX4%
3784 \luatex-def
3785 \MT@requires@luatex3%
3786 \MT@warning{Non-automatic font expansion doesn't work with\MessageBreak
3787 luatex}}%
3788 \luatex-def
3789 { \MT@glet@nc{MT@ex@c@MT@curr@set@name @auto}\@empty}%
3790 \pdfTeX-def \relax
3791 \fi
3792 }

```

Tracking: Interword spacing and outer kerning. The variant with space just in case `\SetTracking` is called inside an argument (e.g., to `\IfFileExists`).

```

3793 \MT@define@opt@key{tr}{spacing}
3794 \MT@define@opt@key{tr}{outerspacing}
3795 \MT@define@opt@key{tr}{outerkerning}

```

Which ligatures should be disabled?

```

3796 \define@key{MT@tr@c}{noligatures}[]{}%
3797 { \MT@xdef@n{MT@tr@c@MT@curr@set@name @noligatures}{#1}}
3798 \define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
3799 \define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
3800 \define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{noligatures={#1}}}
3801 \pdfTeX-def|luatex-def

```

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.

```
3802 <*package>
3803 \renewcommand*\DeclareCharacterInheritance[1][]{%
3804   \let\MT@extra@context\@empty
3805   \let\MT@extra@inputenc\@undefined
3806   \let\MT@inh@feat\@empty
3807   \setkeys{MT@inh@}{#1}%
3808   \MT@begin@catcodes
3809   \MT@set@inh@list
3810 }
```

`\MT@set@inh@list` No need to create an inheritance list for tracking.

```
3811 \def\MT@set@inh@list#1#2{%
3812   \MT@ifempty\MT@inh@feat{%
3813     \MT@map@clist@c\MT@features{%
3814       \MT@ifstreq{#1}{tr}\relax{\MT@declare@char@inh{#1}{#1}{#2}}%
3815     }%
3816   }%
3817   \MT@map@clist@c\MT@inh@feat{%
3818     \KV@esp@def\@tempa{#1}%
3819     \MT@ifempty\@tempa\relax{%
3820       \edef\@tempa{\csname MT@rbba@\@tempa\endcsname}%
3821       \MT@ifstreq\@tempa{tr}\relax{%
3822         \MT@exp@one@n\MT@declare@char@inh{\@tempa}{#1}{#2}}%
3823       }%
3824     }%
3825   \MT@end@catcodes
3826 }
```

The keys for the optional argument.

```
3827 \MT@map@clist@c\MT@features@long{%
3828   \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}
3829   \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
```

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

```
3830 \def\MT@declare@char@inh#1#2#3{%
3831   \MT@edef@n{MT@#1@inh@name}%
3832   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3833   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3834   \MT@ifdefined@c@T\MT@extra@inputenc{%
3835     \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3836   <debug>\MT@info{1}{creating inheritance list ~\@nameuse{MT@#1@inh@name}}%
3837   \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
3838   \def\MT@permutelist{#1@inh}%
3839   \setkeys{MT@inh}{#2}%
3840   \MT@permute
3841 }
```

Parse the second argument. `\DeclareCharacterInheritance` may also be set up for various combinations. We can reuse the key setup from the configuration lists (`\Set...`).

```
3842 \MT@define@code@key{encoding}{inh}
```



```

3843 \MT@define@code@key@family {inh}
3844 \MT@define@code@key{series} {inh}
3845 \MT@define@code@key{shape} {inh}
3846 \MT@define@code@key@size {inh}
3847 \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).

```

3848 \def\MT@inh@do#1,{%
3849   \ifx\relax#1\@empty \else
3850     \MT@inh@split #1==\relax
3851     \expandafter\MT@inh@do
3852   \fi
3853 }

```

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

```

3854 \<package>
3855 \<pdfTeX-def|xetex-def|luatex-def>
3856 \def\MT@inh@split#1=#2=#3\relax{%
3857   \def\@tempa{#1}%
3858   \ifx\@tempa\@empty \else
3859     \MT@get@slot
3860     \<pdfTeX-def|luatex-def> \ifnum\MT@char > \m@ne
3861     \<xetex-def> \ifx\MT@char\@empty\else
3862       \let\MT@val\MT@char
3863       \MT@map@c@list@n{#2}{%
3864         \def\@tempa{##1}%
3865         \ifx\@tempa\@empty \else
3866           \MT@get@slot
3867           \<pdfTeX-def|luatex-def> \ifnum\MT@char > \m@ne
3868           \<xetex-def> \ifx\MT@char\@empty\else
3869             \MT@exp@cs\MT@xadd{\MT@inh@{\MT@listname @\MT@val @}{\MT@char}}%
3870             \fi
3871           \fi
3872         }%
3873       \<debug>\MT@info{n}{2}{children of #1 (\MT@val):
3874       \<debug> \@nameuse{\MT@inh@{\MT@listname @\MT@val @}}%
3875       \fi
3876     \fi
3877   }
3878 \<pdfTeX-def|xetex-def|luatex-def>

```

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. So that,

```

\SetProtrusion
{ encoding = U,
  family   = {euroitc,euroitcs} }
{ E = {100,50} }
\SetProtrusion
{ encoding = U,
  family   = {euroitc,euroitcs},
  shape     = it* }

```

```
{ E = {100,} }
```

would yield the following assignments:

```
3879 (*package)
3880 \MT@gdefen{MT@pr@c@U/euroitc///}{euroitc}
3881 \MT@gdefen{MT@pr@c@U/euroitcs///}{euroitc}
3882 \MT@gdefen{MT@pr@c@U/euroitc//it/}{euroitci}
3883 \MT@gdefen{MT@pr@c@U/euroitcs//it/}{euroitci}
3884 \MT@gdefen{MT@pr@c@euroitc}{E={100,50}}
3885 \MT@gdefen{MT@pr@c@euroitci}{E={100,}}
3886 \def\MT@permute{%
3887   \let\MT@cnt@encoding\@ne
3888   \MT@permute@
```

Undefine commands for the next round.

```
3889 \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
3890 \MT@gllet\MT@tempsize\undefined
3891 }
3892 \def\MT@permute@{%
3893   \let\MT@cnt@family\@ne
3894   \MT@permute@@
3895   \MT@increment\MT@cnt@encoding
3896   \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
3897   \MT@permute@
3898 }
3899 \def\MT@permute@@{%
3900   \let\MT@cnt@series\@ne
3901   \MT@permute@@@
3902   \MT@increment\MT@cnt@family
3903   \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
3904   \MT@permute@@
3905 }
3906 \def\MT@permute@@@{%
3907   \let\MT@cnt@shape\@ne
3908   \MT@permute@@@@
3909   \MT@increment\MT@cnt@series
3910   \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
3911   \MT@permute@@@@
3912 }
3913 \def\MT@permute@@@@{%
3914   \MT@permute@@@@@
3915   \MT@increment\MT@cnt@shape
3916   \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
3917   \MT@permute@@@@@
3918 }
```

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

```
3919 \def\MT@permute@@@@@{%
3920   \MT@permute@define(encoding)%
3921   \ifMT@document
3922     \ifx\MT@tempencoding\empty \else
3923       \MT@ifdefined@n@TF{T@MT@tempencoding}\relax
3924       {\expandafter\expandafter\expandafter\@gobble}%
3925     \fi
3926   \fi
3927   \MT@permute@@@@@@
3928 }
```

\MT@permute@@@@@@

```
3929 \def\MT@permute@@@@@@{%
3930   \MT@permute@define{family}%
3931   \MT@permute@define{series}%
3932   \MT@permute@define{shape}%
3933   \edef\@tempa{\MT@tempencoding
```

```

3934      /\MT@tempfamily
3935      /\MT@tempseries
3936      /\MT@tempshape
3937      /\MT@ifdefined@c@T\MT@tempsize *}%

```

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

```

3938 \MT@ifstreq\@tempa{////}\relax{%
3939   \ifx\MT@tempencoding\empty
3940     \MT@warning{%
3941       You have to specify an encoding for\MessageBreak
3942       \@nameuse{MT@abbr@MT@permutelist} list
3943       ~\@nameuse{MT@MT@permutelist @name}'.\MessageBreak
3944       Ignoring it}%
3945   \else
3946     \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

3947     \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3948       \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3949     }%
3950     \MT@exp@cs\MT@xaddb
3951     {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3952     \MT@tempsize
3953 (debug) \MT@info@n{1}{initialising: use list for font \@tempa,\MessageBreak
3954 (debug)       sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3955 (debug)       @sizes\endcsname}%
3956   }%

```

Only one list can apply to a given combination. But we don't warn if the overridden list is to be loaded by the current one.

```

3957     \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3958       \MT@ifstreq{\csname MT@MT@permutelist @\@tempa\MT@extra@context\endcsname}%
3959       {\csname MT@MT@permutelist @\csname MT@MT@permutelist @name\endcsname @load\endcsname}%
3960       \relax{%
3961         \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3962           ~\@nameuse{MT@MT@permutelist @name}' will\MessageBreak override
3963           list ~\@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3964           for \MessageBreak font ~\@tempa'}%
3965       }%
3966     }%
3967 (debug) \MT@info@n{1}{initialising: use list for font \@tempa
3968 (debug)       \ifx\MT@extra@context\empty\else\MessageBreak
3969 (debug)       (context: \MT@extra@context)\fi}%
3970   }%
3971   \MT@xdef@n{MT@MT@permutelist @\@tempa\MT@extra@context}%
3972   {\csname MT@MT@permutelist @name\endcsname}%
3973   \fi
3974 }%
3975 }

```

\MT@permute@define Define the commands.

```

3976 \def\MT@permute@define#1{%
3977   \@tempcnta=\csname MT@cnt@#1\endcsname\relax
3978   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3979   {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3980   {\MT@let@nc{MT@temp#1}\empty}%
3981 }

```

\MT@permute@reset Reset the commands.

```

3982 \def\MT@permute@reset#1{%
3983   \@tempcnta=\@ne
3984   \MT@loop
3985   \MT@let@nc{MT@temp#1\the\@tempcnta}\undefined

```

```

3986     \advance\@tempcnta\@ne
3987     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3988     \iftrue
3989     \iffalse
3990     \MT@repeat
3991 }

\MT@check@rlist    For every new range item in \MT@tempsize, check whether it overlaps with ranges
                    in the existing list.
3992 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

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

3993 \def\MT@check@rlist@#1#2#3{%
3994     \def\@tempb{#1}%
3995     \def\@tempc{#2}%
3996     \MT@if@false
3997     \MT@exp@cs\MT@map@tlist@c
3998     {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3999     \MT@check@range
4000 }

\MT@check@range    ... recurse through the list of existing ranges.
4001 \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, <#1> and <#2>
                    those of the existing range. <#3> is the list name.
4002 \def\MT@check@range@#1#2#3{%
4003     \MT@ifdim{#2}=\m@ne{%
4004         \MT@ifdim\@tempc=\m@ne{%

            • Both items are simple sizes.

4005         \MT@ifdim\@tempb={#1}\MT@if@true\relax
4006         }{%

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

4007         \MT@ifdim\@tempb>{#1}\relax{%
4008             \MT@ifdim\@tempc>{#1}{%
4009                 \MT@if@true
4010                 \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
4011             }\relax
4012         }%
4013     }%
4014     }{%
4015         \MT@ifdim\@tempc=\m@ne{%

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

4016         \MT@ifdim\@tempb<{#2}{%
4017             \MT@ifdim\@tempb<{#1}\relax\MT@if@true
4018             }\relax
4019         }{%

            • Both items are ranges.

4020         \MT@ifdim\@tempb<{#2}{%
4021             \MT@ifdim\@tempc>{#1}{%
4022                 \MT@if@true
4023                 \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
4024             }\relax
4025         }\relax
4026     }%
4027 }%
4028 \ifMT@if@

```

```

4029 \MT@ifstreq{#3}%
4030 {\csname MT@MT@permutelist @\csname MT@MT@permutelist @name\endcsname @load\endcsname}%
4031 \relax}%
4032 \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
4033 ~\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
4034 list `#3' for font \@tempa,\MessageBreak size \@tempb}%
4035 }%

```

If we've already found a conflict with this item, we can skip the rest of the list.

```

4036 \expandafter\MT@tlist@break
4037 \fi
4038 }

```

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 4039 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 4040 \newif\ifMT@opt@auto
4041 \newif\ifMT@opt@DVI

```

`\MT@optwarn@admissible` Some warnings.

```

4042 \def\MT@optwarn@admissible#1#2{%
4043 \MT@warning@n1{`#1' is not an admissible value for option\MessageBreak
4044 ~`#2'. Assuming `false'}%
4045 }

```

`\MT@optwarn@nan`

```

4046 /package
4047 <*package|letterspace>
4048 plain\MT@requires@latex1{
4049 \def\MT@optwarn@nan#1#2{%
4050 \MT@warning@n1{Value `#1' for option `#2' is not a\MessageBreak number.
4051 ~~~~~Using default value of \number\@nameuse{MT@#2@default}}%
4052 }
4053 plain}\relax
4054 /package|letterspace
4055 *package

```

`\MT@opt@def@set`

```

4056 \def\MT@opt@def@set#1{%
4057 \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
4058 \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
4059 }{%
4060 \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
4061 \MT@warning@n1{The #1 set ~\MT@val' is undeclared.\MessageBreak
4062 ~~~~~Using set ~\@nameuse{MT@\@tempb @setname}' instead}%
4063 }%
4064 }

```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a *<set name>*.

```

4065 \MT@map@clist@n{protrusion,expansion}{%
4066 \define@key{MT}{#1}[true]{%
4067 \csname MT@opt@#1true\endcsname
4068 \MT@map@clist@n{##1}{%
4069 \KV@sp@def\MT@val{###1}%
4070 \MT@ifempty\MT@val\relax{%
4071 \csname MT@#1true\endcsname
4072 \edef\@tempb{\csname MT@rbba@#1\endcsname}%
4073 \MT@ifstreq\MT@val{true}\relax

```

```

4074      {%
4075      \MT@ifstreq\MT@val{false}{%
4076      \csname MT@#1false\endcsname
4077      }{%
4078      \MT@ifstreq\MT@val{compatibility}{%
4079      \MT@let@nc{MT@\@tempb @level}\@ne
4080      }{%
4081      \MT@ifstreq\MT@val{nocompatibility}{%
4082      \MT@let@nc{MT@\@tempb @level}\tw@
4083      }{%

```

If everything failed, it should be a set name.

```

4084      \MT@opt@def@set{#1}%
4085      }%
4086      }%
4087      }%
4088      }%
4089      }%
4090      }%
4091      }%
4092 }

```

activate is a shortcut for protrusion and expansion.

```

4093 \define@key{MT}{activate}[true]{%
4094   \setkeys{MT}{protrusion={#1}}%
4095   \setkeys{MT}{expansion={#1}}%
4096 }

```

spacing, kerning and tracking do not have a compatibility level.

```

4097 \MT@map@clist@n{spacing,kerning,tracking}{%
4098   \define@key{MT}{#1}[true]{%
4099     \MT@map@clist@n{##1}{%
4100       \KV@sp@def\MT@val{###1}%
4101       \MT@ifempty\MT@val\relax{%
4102         \csname MT@#1true\endcsname
4103         \MT@ifstreq\MT@val{true}\relax
4104         {%
4105           \MT@ifstreq\MT@val{false}{%
4106           \csname MT@#1false\endcsname
4107           }{%
4108             \edef\@tempb{\csname MT@rbba@#1\endcsname}%
4109             \MT@opt@def@set{#1}%
4110             }%
4111           }%
4112         }%
4113       }%
4114     }%
4115 }

```

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

```

4116 \def\MT@def@bool@opt#1#2{%
4117   \define@key{MT}{#1}[true]{%
4118     \def\@tempa{##1}%
4119     \MT@ifstreq\@tempa{true}\relax{%
4120       \MT@ifstreq\@tempa{false}\relax{%
4121         \MT@optwarn@admissible{##1}{#1}%
4122         \def\@tempa{false}%
4123       }%
4124     }%
4125     #2%
4126   }%
4127 }

```

Boolean options that only set the switch.

```

4128 \MT@map@clist@n{draft,selected,babel}{%
4129   \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
4130 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotruer}

```

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

```

4131 </package>
4132 <*pdftex-def|luatex-def|xetex-def>
4133 <luatex-def>\MT@requires@luatex4{\let\pdfoutput\outputmode}\relax
4134 \MT@def@bool@opt{DVIoutput}{%
4135   \csname if\@tempa\endcsname
4136   <*pdftex-def|luatex-def>
4137     \ifnum\pdfoutput>\z@ \MT@opt@DVIttrue \fi
4138     \pdfoutput\z@
4139   \else
4140     \ifnum\pdfoutput<\@ne \MT@opt@DVIttrue \fi
4141     \pdfoutput\@ne
4142   </pdftex-def|luatex-def>
4143   <xetex-def> \MT@warning@n{Ignoring `DVIoutput' option}%
4144   \fi
4145 }
4146 </pdftex-def|luatex-def|xetex-def>

```

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.

```

4147 <*package>
4148 \MT@def@bool@opt{defersetup}{%
4149   \csname if\@tempa\endcsname \else
4150     \AtEndOfPackage{%
4151       \MT@setup@
4152       \let\MT@setup@\@empty
4153       \let\MT@addto@setup\@firstofone
4154     }%
4155   \fi
4156 }
4157 </package>

```

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. It also works with LuaT_EX 0.30 or newer.

```

4158 <*pdftex-def|luatex-def>
4159 <pdftex-def>\MT@requires@pdftex7{
4160   \MT@def@bool@opt{copyfonts}{%
4161     \csname if\@tempa\endcsname
4162     \MT@gllet\MT@copy@font\MT@copy@font@
4163   \else
4164     \MT@gllet\MT@copy@font\relax
4165   \fi
4166 }
4167 <pdftex-def>}{
4168 </pdftex-def|luatex-def>
4169 <*pdftex-def|xetex-def>
4170 \MT@def@bool@opt{copyfonts}{%
4171   \csname if\@tempa\endcsname
4172   \MT@error
4173 <pdftex-def> {The pdftex version you are using is too old\MessageBreak
4174 <pdftex-def> to use the `copyfonts' option}{Upgrade pdftex.}%

```

```

4175 <xetex-def>          {The `copyfonts' option does not work with xetex}
4176 <xetex-def>          {Use pdftex or luatex instead.}%
4177   \fi
4178 }
4179 <pdftex-def>{}
4180 </pdftex-def|xetex-def>

```

final is the opposite to draft.

```

4181 <*package>
4182 \MT@def@bool@opt{final}{%
4183   \csname if\@tempa\endcsname
4184   \MT@draftfalse
4185   \else
4186   \MT@drafttrue
4187   \fi
4188 }

```

For verbose output, we redefine \MT@vinfo.

```

4189 \define@key{MT}{verbose}[true]{%
4190   \let\MT@vinfo\MT@info@n1
4191   \def\@tempa{#1}%
4192   \MT@ifstreq\@tempa{true}\relax{%

```

Take problems seriously.

```

4193   \MT@ifstreq\@tempa{errors}{%
4194     \let\MT@warning \MT@warn@err
4195     \let\MT@warning@n1\MT@warn@err
4196   }{%
4197     \let\MT@vinfo\@gobble

```

Cast warnings to the winds.

```

4198   \MT@ifstreq\@tempa{silent}{%
4199     \let\MT@warning \MT@info
4200     \let\MT@warning@n1\MT@info@n1
4201   }{%
4202     \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
4203   }%
4204 }%
4205 }%
4206 }
4207 </package>

```

Options with numerical keys: factor, stretch, shrink, step, letterspace.

```

4208 <*package|letterspace>
4209 <plain>\MT@requires@latex1{
4210 \MT@map@clist@n{%
4211 <package> stretch,shrink,step,%
4212 letterspace}{%
4213 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
4214 \def\@tempa{##1 }%

```

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

```

4215   \MT@ifint\@tempa
4216   {\MT@edef@n{MT@#1}{\@tempa}}%
4217   {\MT@optwarn@nan{##1}{#1}}%
4218 }%
4219 }
4220 <plain>}\relax
4221 </package|letterspace>

```

factor will define the protrusion factor only.

```

4222 <*package>
4223 \define@key{MT}{factor}[\MT@factor@default]{%
4224 \def\@tempa{#1 }%
4225 \MT@ifint\@tempa

```



```

4226 {\edef\MT@pr@factor{\@tempa}}
4227 {\MT@optwarn@nan{#1}{factor}}}%
4228 }

```

Unit for protrusion codes.

```

4229 \define@key{MT}{unit}[character]{%
4230   \def\@tempa{#1}%
4231   \MT@ifstreq\@tempa{character}\relax{%
4232     \MT@ifdimen\@tempa
4233     {\let\MT@pr@unit\@tempa}%
4234     {\MT@warning@n1{\@tempa' is not a dimension.\MessageBreak
4235       Ignoring it and setting values relative to\MessageBreak
4236       character widths}}}%
4237 }%
4238 }

```

14.4.2 Loading the definition file

`\MT@endinput` Abort if no capable engine found.

```

4239 \let\MT@endinput\relax
4240 \ifx\MT@engine\relax
4241   \MT@warning@n1{You don't seem to be using pdfTeX, luatex or xetex.\MessageBreak
4242     ~\MT@MT' only works with these engines.\MessageBreak
4243     I will quit now}
4244   \MT@clear@options
4245 \else

```

Otherwise load the engine-specific code (as strewn across this file).

```

4246   \input{microtype-\MT@engine tex.def}
4247 \fi
4248 \MT@endinput

```

14.4.3 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 modern \TeX systems have switched 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 .)

```

4249 \MT@protrusiontrue
4250 </package>
4251 <*<pdfTeX-def|<luatex-def
4252 \ifnum\pdfoutput<\@ne \else

```

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

```

4253 <pdfTeX-def> \MT@requires@pdfTeX4{
4254   \MT@expansiontrue
4255 <pdfTeX-def> \MT@autottrue
4256 <pdfTeX-def> }\relax
4257 \fi
4258 <luatex-def>\MT@autottrue
4259 <pdfTeX-def|<luatex-def>

```

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.

```

4260 <*<package>
4261 \define@key{MT}{config}[]{\relax}
4262 \def\MT@get@config#1config=#2,#3\@nil{%

```

```

4263 \MT@ifempty{#2}%
4264 {\def\MT@config@file{\MT@MT.cfg}}%
4265 {\def\MT@config@file{#2.cfg}}%
4266 }
4267 \expandafter\expandafter\expandafter\MT@get@config
4268 \csname opt@\currname.\@currentx\endcsname,config=\@nil

```

Load the file.

```

4269 \IfFileExists{\MT@config@file}{%
4270 \MT@info@nl{Loading configuration file \MT@config@file}%
4271 \MT@begin@catcodes
4272 \let\MT@begin@catcodes\relax
4273 \let\MT@end@catcodes\relax
4274 \let\MT@curr@file\MT@config@file
4275 \input{\MT@config@file}%
4276 \endgroup
4277 }{\MT@warning@nl{%
4278 Could not find configuration file '\MT@config@file'!\MessageBreak
4279 This will almost certainly cause undesired results.\MessageBreak
4280 Please fix your installation}%
4281 }

```

`\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).

```

4282 \def\MT@check@active@set#1{%
4283 \MT@ifdefined@n@TF{MT@#1@setname}{%
4284 \MT@info@nl{Using \nameuse{MT@abbr@#1} set \nameuse{MT@#1@setname}'}%
4285 }{%
4286 \MT@ifdefined@n@TF{MT@default@#1@set}{%
4287 \MT@glet@nn{MT@#1@setname}{MT@default@#1@set}%
4288 \MT@info@nl{Using default \nameuse{MT@abbr@#1} set \nameuse{MT@#1@setname}'}%
4289 }{%

```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set '@', and issue a warning.

```

4290 \MT@gdef@n{MT@#1@setname}{@}%
4291 \MT@warning@nl{No \nameuse{MT@abbr@#1} set chosen, no default set declared.
4292 \MessageBreak Using empty set}%
4293 }%
4294 }%
4295 }

```

14.4.4 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.

```

4296 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
4297   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
4298   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
4299 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook

```

14.4.5 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.

```

4300 \def\microtypesetup{\setkeys{MT}}
4301 \MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
4302 \end{package}
4303 \ifx\pdftex-def\luatex-def\ifx\pdftex-def\pdftex-def
4304 \def\MT@define@optionX#1#2{%
4305   \define@key{MTX}{#1}[true]{%
4306     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
4307     \MT@map@cliston{#1}{%
4308       \KV@sp@def\MT@val{###1}%
4309       \MT@ifempty\MT@val\relax{%
4310         \@tempcnta=\m@ne
4311       }

```

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.

```

4312   \MT@checksetup{#1}{%
4313     \@tempcnta=\csname MT@\@tempb @level\endcsname
4314     \MT@vinfo{Enabling #1
4315       (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
4316   }%
4317 }{%
4318   \MT@ifstreq\MT@val{false}{%
4319     \@tempcnta=\z@
4320     \MT@vinfo{Disabling #1\on@line}%
4321   }{%
4322     \MT@ifstreq\MT@val{compatibility}{%
4323       \MT@checksetup{#1}{%
4324         \@tempcnta=\@ne
4325         \MT@let@nc{MT@\@tempb @level}\@ne
4326         \MT@vinfo{Setting #1 to level 1\on@line}%
4327       }%
4328     }{%
4329       \MT@ifstreq\MT@val{nocompatibility}{%
4330         \MT@checksetup{#1}{%
4331           \@tempcnta=\tw@
4332           \MT@let@nc{MT@\@tempb @level}\tw@
4333           \MT@vinfo{Setting #1 to level 2\on@line}%
4334         }%
4335       }{\MT@error{Value '\MT@val' for key '#1' not recognised}}

```

```

4336                                     {Use any of `true', `false', `compatibility' or
4337                                     `nocompatibility'.}%
4338                                     }%
4339                                 }%
4340                            }%
4341                       }%
4342                \ifnum\@tempcnta>\m@ne
4343                #2\@tempcnta\relax
4344                \fi
4345            }%
4346        }%
4347    }%
4348 }

```

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

```

4349 \def\MT@checksetup#1{%
4350   \csname ifMT@#1\endcsname
4351   \expandafter\@firstofone
4352   \else
4353     \MT@error{You cannot enable #1 if it was disabled\MessageBreak
4354             in the package options}{Load microtype with #1 enabled.}%
4355     \expandafter\@gobble
4356   \fi
4357 }

4358 \MT@define@optionX{protrusion}\MT@protrudechars
4359 (/pdfTeX-def|luaTeX-def|xetex-def)
4360 (*pdfTeX-def|luaTeX-def)
4361 \MT@define@optionX{expansion}\MT@adjustspacing

```

`\MT@protrudechars`

```

\MT@adjustspacing 4362 (*luaTeX-def)
4363 \MT@requires@luaTeX4{
4364   \let\pdfprotrudechars\protrudechars
4365   \let\pdfadjustspacing\adjustspacing
4366 } \relax
4367 (/luaTeX-def)
4368 \let\MT@protrudechars\pdfprotrudechars
4369 \let\MT@adjustspacing\pdfadjustspacing
4370 (/pdfTeX-def|luaTeX-def)
4371 (*xetex-def)
4372 \let\MT@protrudechars\XeTeXprotrudechars
4373 \define@key{MTX}{expansion}[true]{\MT@warning{Ignoring expansion setup}}
4374 (/xetex-def)

```

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

```

4375 (*pdfTeX-def|luaTeX-def)
4376 (pdfTeX-def) \MT@requires@pdfTeX6{
4377 (luaTeX-def) \MT@requires@luaTeX3{
4378   \def\MT@define@optionX@#1#2{%
4379     \define@key{MTX}{#1}[true]{%
4380       \MT@map@clist@n{##1}{%
4381         \KV@sp@def\MT@val{###1}%
4382         \MT@ifempty\MT@val\relax{%
4383           \@tempcnta=\m@ne
4384           \MT@ifstreq\MT@val{true}{%
4385             \MT@checksetup{#1}{%
4386               \@tempcnta=\@ne
4387               \MT@vinfo{Enabling #1\on@line}%
4388             }%
4389           }{%
4390             \MT@ifstreq\MT@val{false}{%
4391               \@tempcnta=\z@
4392               \MT@vinfo{Disabling #1\on@line}%

```

```

4393         {\MT@error{Value `~\MT@val' for key `~#1' not recognised}
4394         {Use either `true' or `false'}}%
4395     }%
4396 }%
4397 \ifnum\@tempcnta>\m@ne
4398     #2\relax
4399 \fi
4400 }%
4401 }%
4402 }%
4403 }

```

We cannot simply let `\MT@tracking` relax, since this may select the already letter-spaced font instance.

```

4404 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
4405                               \else \let\MT@tracking\MT@tracking@ \fi}
4406 (pdfTeX-def) \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
4407 (pdfTeX-def) \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
4408 (pdfTeX-def) \pdfappendkern\@tempcnta}
4409 }{
4410 (pdfTeX-def|LaTeX-def)
4411 (pdfTeX-def|LaTeX-def|xTeX-def)

```

Disable for older pdfTeX versions and for XeTeX and LuaTeX.

```

4412 \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
4413 (LaTeX-def)
4414 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
4415 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
4416 (pdfTeX-def)
4417 \define@key{MTX}{activate}[true]{%
4418     \setkeys{MTX}{protrusion={#1}}}%
4419 (pdfTeX-def|LaTeX-def) \setkeys{MTX}{expansion={#1}}}%
4420 }
4421 (pdfTeX-def|LaTeX-def|xTeX-def)

```

`\MT@saved@setupfont` Disable everything – may be used as a temporary work-around in case setting up fonts doesn't work under certain circumstances, but only until that specific problem is fixed. This is *undocumented*, as it completely deprives us of the possibility to act – we're blind and paralysed.

```

4422 (package)
4423 \let\MT@saved@setupfont\MT@setupfont
4424 \define@key{MTX}{disable}[]{}%
4425 \MT@info{Inactivate `~\MT@MT' package}%
4426 \let\MT@setupfont\relax
4427 }
4428 \define@key{MTX}{enable}[]{}%
4429 \MT@info{Reactivate `~\MT@MT' package}%
4430 \let\MT@setupfont\MT@saved@setupfont
4431 }
4432 (package)

```

14.4.6 Processing the options

`\MT@ProcessOptionsWithKV` Parse options.

```

4433 (package|letterspace)
4434 (plain) \MT@requires@latex1{
4435 \def\MT@ProcessOptionsWithKV#1{%
4436     \let\@tempc\relax
4437     \let\MT@temp\@empty
4438 (plain) \MT@requires@latex2{
4439     \MT@map@clist@c\@classoptionslist{%
4440     \def\CurrentOption{##1}%

```

```

4441 \MT@ifdefined@n@T{KV@#1@}\expandafter\MT@getkey\CurrentOption=\@nil}{%
4442 \edef\MT@temp{\MT@temp,\CurrentOption,}%
4443 \@expandtwoargs\@removeelement\CurrentOption
4444 \@unusedoptionlist\@unusedoptionlist
4445 }%
4446 }%
4447 \edef\MT@temp{\noexpand\setkeys{#1}%
4448 {\MT@temp\@optionlist{\@currname.\@currentext}}}%

```

plain can handle package options.

```

4449 <plain>
4450 }{\edef\MT@temp{\noexpand\setkeys{#1}%
4451 {\csname usepkg@options@usepkg@pkg\endcsname}}}%
4452 </plain>
4453 \MT@temp
4454 \MT@clear@options
4455 }

```

\MT@getkey For key=val in class options.

```

4456 \def\MT@getkey#1=#2\@nil{#1}
4457 \MT@ProcessOptionsWithKV{MT}
4458 <plain>)\relax
4459 </package|letterspace>
4460 <*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).

```

4461 \MT@addto@setup{%
4462 \ifMT@draft

```

We disable most of what we've just defined in the 4462 lines above if we are running in draft mode.

```

4463 \MT@warning@n1{\`draft' option active.\MessageBreak
4464 Disabling all micro-typographic extensions.\MessageBreak
4465 This might lead to different line and page breaks}%
4466 \let\MT@setupfont\relax
4467 \renewcommand*\LoadMicrotypeFile[1]{}%
4468 \renewcommand*\microtypesetup[1]{}%
4469 \renewcommand*\microtypecontext[1]{}%
4470 \renewcommand*\lsstyle{}%
4471 \else
4472 \MT@setup@PDF
4473 \MT@setup@copies

```

Fix the font sets.

```

4474 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
4475 \MT@setup@protrusion
4476 \MT@setup@expansion
4477 \MT@setup@tracking
4478 \MT@setup@warntracking
4479 \MT@setup@spacing
4480 \MT@setup@kerning
4481 \MT@setup@noligatures
4482 }
4483 </package>

```

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

Some packages depend on the value of \pdfoutput and will get confused if it is changed after they have been loaded. These packages are, among others: color, graphics, hyperref, crop, contour, pstricks and, as a matter of course, ifpdf.

Instead of testing for each package (that's not our job), we only say that it was microtype that changed it. This must be sufficient!

```

4484 (*pdfTeX-def|luatex-def)
4485 \def\MT@setup@PDF{%
4486   \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
4487     \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
4488 }

```

\MT@setup@copies Working on font copies?

```

4489 \def\MT@setup@copies{%
4490   \ifx\MT@copy@font\relax\else \MT@info@nl{Using font copies for contexts}\fi
4491 }
4492 (/pdfTeX-def|luatex-def)
4493 (*xetex-def)
4494 \let\MT@setup@PDF\relax
4495 \let\MT@setup@copies\relax
4496 (/xetex-def)

```

\MT@setup@protrusion Protrusion.

```

4497 (*pdfTeX-def|xetex-def|luatex-def)
4498 \def\MT@setup@protrusion{%
4499   \ifMT@protrusion
4500     \edef\MT@active@features{\MT@active@features,pr}%
4501     \MT@protrudechars\MT@pr@level
4502     \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
4503       \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
4504         factor: \number\MT@pr@factor\fi
4505       \ifx\MT@pr@unit@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
4506     \MT@check@active@set{pr}%
4507   \else
4508     \let\MT@protrusion\relax
4509     \MT@info@nl{No character protrusion}%
4510   \fi
4511 }
4512 (/pdfTeX-def|xetex-def|luatex-def)

```

\MT@setup@expansion For DVI output, the user must have explicitly passed the expansion option to the package. Under LuaTeX, expansion works quite differently: the glyphs will be positioned as if they were transformed, without actually being transformed. Since this could still be considered a viable option, we don't disable the feature completely, but issue a warning.

```

4513 (*pdfTeX-def|luatex-def)
4514 \def\MT@setup@expansion{%
4515   \ifnum\pdfoutput<\@ne
4516     \ifMT@opt@expansion
4517       (*luatex-def)
4518       \ifMT@expansion
4519         \MT@requires@luatex3{%
4520           \MT@warning@nl{Font expansion doesn't work properly with luatex in\MessageBreak
4521             DVI mode: the glyphs won't be actually transformed,\MessageBreak
4522             but will only be shifted. You might want to use\MessageBreak
4523             pdfLaTeX instead. I'll continue anyway ..}%
4524           %\MT@expansionfalse
4525         }\relax
4526       \fi
4527     (/luatex-def)
4528   \else
4529     \MT@expansionfalse
4530   \fi
4531 \fi
4532 \ifMT@expansion

```

Set up the values for font expansion: if stretch has not been specified, we take the

default value of 20.

```
4533 \ifnum\MT@stretch=\m@ne
4534 \let\MT@stretch\MT@stretch@default
4535 \fi
```

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

```
4536 \ifnum\MT@shrink=\m@ne
4537 \let\MT@shrink\MT@stretch
4538 \fi
```

If step has not been specified, we will just set it to 1 for recent pdfTeX versions. My tests did not show much difference neither in compilation time (within the margin of error) nor in file size (less than 1% difference for microtype.pdf with step=1 compared to step=5). With older versions, we set it to min(stretch,shrink)/5, rounded off, minimum value 1.

```
4539 \ifnum\MT@step=\m@ne
4540 <pdfTeX-def> \MT@requires@pdftex6{%
4541 \def\MT@step{1}%
4542 <pdfTeX-def>
4543 }{%
4544 \ifnum\MT@stretch>\MT@shrink
4545 \ifnum\MT@shrink=\z@
4546 \@tempcnta=\MT@stretch
4547 \else
4548 \@tempcnta=\MT@shrink
4549 \fi
4550 \else
4551 \ifnum\MT@stretch=\z@
4552 \@tempcnta=\MT@shrink
4553 \else
4554 \@tempcnta=\MT@stretch
4555 \fi
4556 \fi
4557 \divide\@tempcnta 5\relax
4558 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
4559 \edef\MT@step{\number\@tempcnta\space}%
4560 }%
4561 </pdfTeX-def>
4562 \fi
4563 \ifnum\MT@step=\z@
4564 \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
4565 Setting it to one}%
4566 \def\MT@step{1}%
4567 \fi
```

`\MT@auto` Automatic expansion of the font? This new feature of pdfTeX 1.20 makes the *hz* programme really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdfTeX). With LuaTeX, we just leave it empty, as there’s actually no difference – non-automatic font expansion doesn’t work anymore. In LuaTeX 1.0.6, the ‘autoexpand’ option seems to have been removed altogether and would trigger a warning.

```
4568 \let\MT@auto\@empty
4569 \ifMT@auto
```

We turn off automatic expansion if output mode is DVI.

```
4570 <pdfTeX-def>
4571 \MT@requires@pdftex4{%
4572 \ifnum\pdfoutput<\@ne
4573 \ifMT@opt@auto
4574 \MT@error{%
4575 Automatic font expansion only works for PDF output.\MessageBreak
4576 However, you are creating a DVI file}
```



```

4577         {If you have created expanded fonts instances, remove `auto' from%
4578         \MessageBreak the package options. Otherwise, you have to switch
4579         off expansion\MessageBreak completely.}%
4580     \fi
4581     \MT@autofalse
4582 \else
4583     \def\MT@auto{autoexpand}%
4584 \fi

```

Also, if pdfTeX is too old.

```

4585     }{%
4586     \MT@error{%
4587     The pdftex version you are using is too old for\MessageBreak
4588     automatic font expansion}%
4589     {If you have created expanded fonts instances, remove `auto' from\MessageBreak
4590     the package options. Otherwise, you have to switch off expansion\MessageBreak
4591     completely, or upgrade pdftex to version 1.20 or newer.}%
4592     \MT@autofalse
4593     \def\MT@auto{1000 }%
4594     }%
4595 \end{pdfTeX-def}
4596 \end{luatex-def} \MT@requires@luatex3\relax{\def\MT@auto{autoexpand}}%
4597 \else
4598 \end{pdfTeX-def}

```

No automatic expansion.

```

4599     \MT@requires@pdftex4\relax{%
4600     \def\MT@auto{1000 }%
4601     }%
4602 \end{pdfTeX-def}
4603 \end{luatex-def}
4604 \MT@requires@luatex3{%
4605     \ifMT@opt@auto
4606     \MT@error{Non-automatic font expansion does not work with\MessageBreak
4607     luatex}{Remove `auto=false' from the package options, or use pdftex.}%
4608     \MT@autotruetrue
4609     \fi
4610     }\relax
4611 \end{luatex-def}
4612 \fi

```

Choose the appropriate macro for selected expansion.

```

4613     \ifMT@selected
4614     \let\MT@set@ex@codes\MT@set@ex@codes@s
4615     \else
4616     \let\MT@set@ex@codes\MT@set@ex@codes@n
4617     \fi

```

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

```

4618     \ifnum\MT@stretch=\z@
4619     \ifnum\MT@shrink=\z@
4620     \MT@warning@nl{%
4621     Both the stretch and shrink limit are set to zero.\MessageBreak
4622     Disabling font expansion}%
4623     \MT@expansionfalse
4624     \fi
4625     \fi
4626 \fi
4627 \ifMT@expansion
4628 \edef\MT@active@features{\MT@active@features,ex}%
4629 \MT@adjustspacing\MT@ex@level
4630 \MT@info@nl{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
4631 (level \number\MT@ex@level),\MessageBreak
4632 stretch: \number\MT@stretch, shrink: \number\MT@shrink,
4633 step: \number\MT@step, \ifMT@selected\else non-\fi selected}%

```

`\MT@check@step` Check whether stretch and shrink are multiples of step.

```

4634 \def\MT@check@step##1{%
4635   \@tempcnta=\csname MT@##1\endcsname
4636   \divide\@tempcnta \MT@step
4637   \multiply\@tempcnta \MT@step
4638   \ifnum\@tempcnta=\csname MT@##1\endcsname\else
4639     \MT@warning@n1{The ##1 amount is not a multiple of step.\MessageBreak
4640       The effective maximum ##1 is \the\@tempcnta\space
4641       (step \number\MT@step)}%
4642   \fi
4643 }%
4644 \MT@check@step{stretch}%
4645 \MT@check@step{shrink}%
4646 \MT@check@active@set{ex}%

```

`\showhyphens` Inside `\showhyphens`, font expansion should be disabled. (Since 2017/01/10, the \LaTeX format contains a different version for $X_{\text{Y}}\TeX$, but since expansion doesn't work with $X_{\text{Y}}\TeX$, we don't have to bother.) Since 2019/10/01, the command is robust.

```

4647 \MT@ifdefined@nTF{showhyphens }{%
4648   \def\MT@temp##1##2{%
4649     \expandafter\CheckCommand\csname showhyphens \endcsname[1]{##1}%
4650     \DeclareRobustCommand\showhyphens[1]{##2}}%
4651 }{%
4652   \def\MT@temp##1##2{%
4653     \CheckCommand*\showhyphens[1]{##1}%
4654     \gdef\showhyphens###1{##2}}%
4655 }%
4656 \MT@temp
4657 { \setbox0\vbox{\color@begingroup
4658   \everypar{}\parfillskip\z@skip
4659   \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
4660   \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}
4661 { \setbox0\vbox{\color@begingroup\pdfadjustspacing\z@
4662   \everypar{}\parfillskip\z@skip
4663   \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
4664   \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}%
4665 \else
4666   \let\MT@expansion\relax
4667   \MT@info@n1{No font expansion}%
4668 \fi
4669 }
4670 /pdftex-def|luatex-def
4671 *xetex-def
4672 \def\MT@setup@expansion{%
4673   \ifMT@expansion
4674     \ifMT@opt@expansion
4675       \MT@error{Font expansion does not work with xetex}
4676       {Use pdftex or luatex instead.}%
4677     \fi
4678   \fi
4679 }
4680 /xetex-def

```

`\MT@setup@tracking` Tracking, spacing and kerning.

```

4681 *pdftex-def|luatex-def
4682 pdftex-def \MT@requires@pdftex6{%
4683 luatex-def \MT@requires@luatex3{%
4684   \def\MT@setup@tracking{%
4685     \ifMT@tracking
4686       \edef\MT@active@features{\MT@active@features,tr}%
4687       \MT@info@n1{Tracking enabled}%
4688       \MT@check@active@set{tr}%

```

Enable protrusion for compensation at the line edges.

```

4689 \ifMT@protrusion\else\MT@protrudechars\@ne\fi
4690 \else
4691 \let\MT@tracking\relax
4692 \MT@info@nl{No adjustment of tracking}%
4693 \fi
4694 }
4695 \pdfTeX-def|luatex-def)

```

\MT@setup@spacing

```

4696 (*pdfTeX-def)
4697 \def\MT@setup@spacing{%
4698 \ifMT@spacing
4699 \edef\MT@active@features{\MT@active@features,sp}%
4700 \pdfadjustinterwordglue\@ne
4701 \MT@info@nl{Adjustment of interword spacing enabled}%

```

The ragged2e package sets interword spaces to a fixed value without glue. microtype's modifications can therefore have undesired effects. Therefore, we issue a warning.

```

4702 \MT@with@package@T{ragged2e}{%
4703 \MT@warning@nl{You are using the `ragged2e' package.\MessageBreak
4704 Adjustment of interword spacing may lead to\MessageBreak
4705 undesired results when used with `ragged2e'.\MessageBreak
4706 In this case, disable the `spacing' option}%
4707 }%
4708 \MT@check@active@set{sp}%
4709 \else
4710 \let\MT@spacing\relax
4711 \MT@info@nl{No adjustment of interword spacing}%
4712 \fi
4713 }

```

\MT@setup@spacing@check

Warning if \nonfrenchspacing is active, since space factors will be ignored with \pdfadjustinterwordglue > 0. Why 1500? Because some packages redefine \frenchspacing.¹⁶

```

4714 \def\MT@setup@spacing@check{%
4715 \ifMT@spacing
4716 \ifMT@babel \else
4717 \ifnum\sfcode\> 1500
4718 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
4719 \MT@warning@nl{%
4720 \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
4721 interword spacing will disable it. You might want\MessageBreak
4722 to add `\'backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
4723 to your preamble}%
4724 }%
4725 \fi
4726 \fi
4727 \fi
4728 }

```

\MT@setup@kerning

```

4729 \def\MT@setup@kerning{%
4730 \ifMT@kerning
4731 \edef\MT@active@features{\MT@active@features,kn}%
4732 \pdfprependkern\@ne
4733 \pdfappendkern\@ne
4734 \MT@info@nl{Adjustment of character kerning enabled}%
4735 \MT@check@active@set{kn}%
4736 \else
4737 \let\MT@kerning\relax

```

16 Cf. the c.t.t. thread ‘\frenchspacing with AMS packages and babel’, started by Philipp Lehman on 16 August 2005, MID: ddtbaj\$rob\$1@online.de

```

4738     \MT@info@nl{No adjustment of character kerning}%
4739     \fi
4740   }
4741 </pdfTeX-def>

\MT@error@doesnt@work    If pdfTeX is too old, we disable tracking, spacing and kerning, and throw an error
                        message. We also switch the features off for LuaTeX and XeTeX.
4742 <pdfTeX-def|luaTeX-def>{}
4743 <*luaTeX-def>
4744   \def\MT@setup@tracking{%
4745     \ifMT@tracking
4746       \MT@error{The tracking feature only works with luatex 0.62\MessageBreak
4747         or newer. Switching it off}{Upgrade luatex.}%
4748       \MT@trackingfalse
4749       \MT@let@nc{MT@tracking}\relax
4750     \else
4751       \MT@info@nl{No adjustment of tracking (luatex too old)}%
4752       \fi
4753   }
4754 }
4755 </luaTeX-def>
4756 <*pdfTeX-def|xetex-def|luaTeX-def>
4757   \def\MT@error@doesnt@work#1{%
4758     \csname ifMT@#1\endcsname
4759     \MT@error{The #1 feature only works with pdfTeX 1.40\MessageBreak
4760       or newer. Switching it off}{Upgrade pdfTeX.}%
4761 <pdfTeX-def>           {Use pdfTeX instead.}%
4762 <luaTeX-def|xetex-def>
4763     \csname MT@#1false\endcsname
4764     \MT@let@nc{MT@#1}\relax
4765   \else
4766     \MT@info@nl{No adjustment of #1%
4767 <pdfTeX-def>           \space(pdfTeX too old)%
4768     }%
4769     \fi
4770   }
4771 <pdfTeX-def|xetex-def> \def\MT@setup@tracking{\MT@error@doesnt@work{tracking}}
4772 \def\MT@setup@kerning {\MT@error@doesnt@work{kerning}}
4773 \def\MT@setup@spacing {\MT@error@doesnt@work{spacing}}
4774 <pdfTeX-def>{}
4775 </pdfTeX-def|xetex-def|luaTeX-def>

\MT@setup@warntracking
4776 <letterspace>\MT@addto@setup
4777 <pdfTeX-def|luaTeX-def>\def\MT@setup@warntracking

\MT@warn@tracking@DVI    With pdfTeX, 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.
4778 <*pdfTeX-def|luaTeX-def|letterspace>
4779 {%
4780 <*pdfTeX-def|letterspace>
4781   \ifnum\pdfoutput<\@one
4782     \def\MT@warn@tracking@DVI{%
4783 <letterspace>           \MT@pdf@or@lua{%
4784       \MT@warning@nl{%
4785         You are using tracking/letterspacing in DVI mode.\MessageBreak
4786         This will probably not work, unless the post-\MessageBreak
4787         processing program (dvips, dvipdfm(x), ...) is\MessageBreak
4788         able to create the virtual fonts on the fly}%
4789 <letterspace>       }\relax
4790       \MT@gl@et\MT@warn@tracking@DVI\relax
4791     }%
4792   \else

```

```

4793 </pdfTeX-def|letterspace>
4794   \def\MT@warn@tracking@DVI{%
4795     \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
4796     \MT@gl@et\MT@warn@tracking@DVI\relax
4797   }%
4798 </pdfTeX-def|letterspace> \fi
4799   \ifnum\MT@letterspace=\m@ne
4800     \let\MT@letterspace\MT@letterspace@default
4801   \else
4802     \MT@ls@too@large\MT@letterspace
4803   \fi
4804 }
4805 </pdfTeX-def|luatex-def|letterspace>
4806 <xetex-def>\let\MT@setup@warn@tracking\relax

```

`\MT@setup@noligatures` `\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```

4807 <*pdfTeX-def|luatex-def>
4808 \def\MT@setup@noligatures{%
4809 <pdfTeX-def> \MT@requires@pdfTeX5{%
4810   \ifMT@noligatures \else
4811     \let\MT@noligatures\relax
4812   \fi
4813 <pdfTeX-def> }\relax
4814 }
4815 </pdfTeX-def|luatex-def>
4816 <xetex-def>\let\MT@setup@noligatures\relax

```

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

```

4817 <*package>
4818 \MT@addto@setup{%
4819   \if\MT@active@features\@empty \else
4820     \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
4821   \fi
4822   \MT@documenttrue
4823 }

```

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

```

4824 \def\MT@set@babel@context#1{%
4825   \MT@ifdefined@n@TF{MT@babel@#1}{%
4826     \MT@info{*** Changing to language context `#1'\MessageBreak\on@line}%
4827     \expandafter\MT@exp@one@n\expandafter\microtypecontext
4828     \csname MT@babel@#1\endcsname
4829   }{%
4830     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
4831   }%
4832 }

```

`\MT@shorthandoff` Active characters can only be switched off if babel isn't loaded after microtype.

```

4833 \ifpackageloaded{babel}{
4834   \def\MT@shorthandoff#1#2{%
4835     \MT@info@n1{Switching off #1 babel's active characters (#2)}%
4836     \shorthandoff{#2}}
4837 }{
4838   \def\MT@shorthandoff#1#2{%
4839     \MT@error{You must load `babel' before `~\MT@MT'}
4840     {Otherwise, `~\MT@MT' cannot switch off #1 babel's\MessageBreak
4841       active characters.}}
4842 }

```

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

```

4843 \MT@addto@setup{%
4844   \ifMT@babel

```

```

4845 \ifpackageloaded{babel}{%
4846 \MT@info@n{Redefining babel's language switching commands}%
4847 \let\MT@orig@select@language\select@language
4848 \def\select@language#1{%
4849 \MT@orig@select@language{#1}%
4850 \MT@set@babel@context{#1}%
4851 }%
4852 \let\MT@orig@foreign@language\foreign@language
4853 \def\foreign@language#1{%
4854 \MT@orig@foreign@language{#1}%
4855 \MT@set@babel@context{#1}%
4856 }%
4857 \ifMT@kerning

```

Disable French babel's active characters.

```

4858 \MT@if@false
4859 \MT@with@babel@and@T{french} \MT@if@true
4860 \MT@with@babel@and@T{frenchb} \MT@if@true
4861 \MT@with@babel@and@T{français} \MT@if@true
4862 \MT@with@babel@and@T{canadien} \MT@if@true
4863 \MT@with@babel@and@T{acadian} \MT@if@true
4864 \ifMT@if@MT@shorthandoff{French}{:;!}\fi

```

Disable Turkish babel's active characters.

```

4865 \MT@if@false
4866 \MT@with@babel@and@T{turkish} \MT@if@true
4867 \ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi
4868 \fi

```

In case babel was loaded before microtype:

```

4869 \MT@set@babel@context\languagename
4870 }{%
4871 \MT@warning@n{You did not load the babel package.\MessageBreak
4872 The `babel' option won't have any effect}%
4873 }%
4874 \fi
4875 }

```

Now we close the \fi from \ifMT@draft.

```

4876 \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.

```

4877 \selectfont}

```

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

```

4878 \edef\MT@curr@file{\jobname.tex}
4879 </package>

```

Finally, execute the setup macro at the end of the preamble, and empty it (the combine class calls it repeatedly).

```

4880 <*package|letterspace>
4881 <plain>\MT@requires@latex1{
4882 \AtBeginDocument{\MT@setup@ \MT@gl@et\MT@setup@\@empty}
4883 <plain>}\relax
4884 </package|letterspace>

```

Must come at the very, very end.

```

4885 <package>\MT@ifdefined@c@T\MT@setup@spacing@check
4886 <package> {\AtBeginDocument{\MT@setup@spacing@check}}

```

Restore catcodes.

```

4887 <package|letterspace>\MT@restore@catcodes

```

That was that.

15 Configuration files

Let's now write the font configuration files.

```
4888 (*config)
4889
```

15.1 Font sets

We first declare some sets in the main configuration file.

```
4890 (*m-t)
4891 %%% -----
4892 %%% FONT SETS
4893
4894 \DeclareMicrotypeSet{all}
4895 { }
4896
4897 \DeclareMicrotypeSet{allmath}
4898 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U} }
4899
4900 \DeclareMicrotypeSet{alltext}
4901 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU} }
4902
4903 \DeclareMicrotypeSet{allmath-nott}
4904 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U},
4905   family   = {rm*,sf*}
4906 }
4907
4908 \DeclareMicrotypeSet{alltext-nott}
4909 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
4910   family   = {rm*,sf*}
4911 }
4912
4913 \DeclareMicrotypeSet{basicmath}
4914 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,OML,OMS},
4915   family   = {rm*,sf*},
4916   series    = {md*},
4917   size      = {normalsize,footnotesize,small,large}
4918 }
4919
4920 \DeclareMicrotypeSet{basictext}
4921 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU},
4922   family   = {rm*,sf*},
4923   series    = {md*},
4924   size      = {normalsize,footnotesize,small,large}
4925 }
4926
4927 \DeclareMicrotypeSet{smallcaps}
4928 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
4929   shape     = {sc*,si,scit}
4930 }
4931
4932 \DeclareMicrotypeSet{footnotesize}
4933 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
4934   size      = {-small}
4935 }
4936
4937 \DeclareMicrotypeSet{scriptsize}
4938 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
```

```

4939     size      = {-footnotesize}
4940   }
4941
4942 \DeclareMicrotypeSet{normal font}
4943   { font = */*/*/*/* }
4944

```

The default sets.

```

4945 %%% -----
4946 %%% DEFAULT SETS
4947
4948 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4949 \DeclareMicrotypeSetDefault[expansion]{alltext-nott}
4950 \DeclareMicrotypeSetDefault[spacing]{alltext-nott}
4951 \DeclareMicrotypeSetDefault[kerning]{alltext}
4952 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
4953

```

15.2 Font variants and aliases

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals): Additionally, we add the now common variants for Lining, Tabular, Oldstyle, and Tabular Oldstyle numbers.

```

4954 %%% -----
4955 %%% FONT VARIANTS AND ALIASES
4956
4957 \DeclareMicrotypeVariants{x,j,w,a,d,0,1,-LF,-TLF,-OsF,-TosF}

```

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 just a variant, i.e., they shouldn't share a file.

Fonts that are 'the same': The fontspec package will set lmr as the default font, whose declarations for EU1/EU2/TU encoding are in mt-LatinModernRoman.cfg. Since 2016/12/03, the default encoding with Xe_{La}TeX and Lua_{La}TeX in the L^AT_EX format is TU, even if fontspec is not loaded.

```

4958
4959 \MT@if@false
4960 \ifx\UnicodeEncodingName\undefined\else
4961   \MT@if@fstreq{\encodingdefault}{\UnicodeEncodingName}\MT@if@true\relax
4962 \fi
4963 \ifMT@fontspec\MT@if@true\fi
4964 \ifMT@if@
4965 % -- Computer/Latin Modern Roman
4966 \DeclareMicrotypeAlias{lmr}{Latin Modern Roman}
4967   \else
4968 \DeclareMicrotypeAlias{lmr}{cmr}           % lmodern
4969 \fi

```

The Latin Modern fonts, the virtual fonts from the ae and zefonts and the eco and hfoldsty packages (oldstyle numerals), as well as mllmodern, all inherit the (basic) settings from Computer Modern Roman. Some of them are in part overwritten later. We mustn't forget the Latin Modern math fonts.

```

4970 \DeclareMicrotypeAlias{lmsy}{cmsy}          % "
4971 \DeclareMicrotypeAlias{lmm}{cmm}           % "
4972 \DeclareMicrotypeAlias{aer}{cmr}           % ae
4973 \DeclareMicrotypeAlias{zer}{cmr}           % zefonts

```



```

4974 \DeclareMicrotypeAlias{cmor}{cmr}          % eco
4975 \DeclareMicrotypeAlias{hfor}{cmr}          % hfoldsty
4976 \DeclareMicrotypeAlias{mlmr}{cmr}          % mlmodern
4977 \DeclareMicrotypeAlias{mlmsy}{cmsy}        % "
4978 \DeclareMicrotypeAlias{mlmm}{cmm}          % "

```

Another, new Computer Modern extension. The newcomputermodern package loads it by file name.

```

4979 \DeclareMicrotypeAlias{New Computer Modern}{Latin Modern Roman}
4980 \DeclareMicrotypeAlias{NewCM10-Book.otf}{Latin Modern Roman}
4981 \DeclareMicrotypeAlias{NewCM10-Regular.otf}{Latin Modern Roman}

```

The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively, also the T_EX Gyre fonts Pagella and Termes (formerly: qfonts).

```

4982 %% -- Palatino
4983 \DeclareMicrotypeAlias{pxr}{ppl}            % pxfonts
4984 \DeclareMicrotypeAlias{qpl}{ppl}            % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

```

The ‘FPL Neu’ fonts, a ‘re-implementation’ of Palatino.

```

4985 \DeclareMicrotypeAlias{fp9x}{pplx}         % FPL Neu
4986 \DeclareMicrotypeAlias{fp9j}{pplj}         % "

```

The newpx package, a replacement for pxfonts.

```

4987 \DeclareMicrotypeAlias{zpllf}{pplx}        % newpxtext
4988 \DeclareMicrotypeAlias{zplosf}{pplj}        % "
4989 \DeclareMicrotypeAlias{zpltlf}{pplx}        % "
4990 \DeclareMicrotypeAlias{zpltosf}{pplj}       % "

```

The domitian package.

```

4991 \DeclareMicrotypeAlias{Domitian-TLF}{pplx}% domitian
4992 \DeclareMicrotypeAlias{Domitian-T0sF}{pplj}% "

```

The OpenType versions:

```

4993 \DeclareMicrotypeAlias{Palatino Linotype}{Palatino}
4994 \DeclareMicrotypeAlias{Palatino LT Std}{Palatino}
4995 \DeclareMicrotypeAlias{TeX Gyre Pagella}{Palatino}
4996 \DeclareMicrotypeAlias{Domitian}{Palatino}
4997 \DeclareMicrotypeAlias{Asana Math}{Palatino}
4998 %% -- Times New Roman
4999 \DeclareMicrotypeAlias{txr}{ptm}            % txfonts

```

The newtx package, a replacement for txfonts.

```

5000 \DeclareMicrotypeAlias{ntxlf}{ptmx}        % newtxtext
5001 \DeclareMicrotypeAlias{ntxosf}{ptmj}        % "
5002 \DeclareMicrotypeAlias{ntxtlf}{ptmx}        % "
5003 \DeclareMicrotypeAlias{ntxtosf}{ptmj}       % "

```

The tempora package.

```

5004 \DeclareMicrotypeAlias{Tempora-TLF}{ptmx} % tempora
5005 \DeclareMicrotypeAlias{Tempora-T0sF}{ptmj} % "
5006 \DeclareMicrotypeAlias{qtm}{ptm}           % TeX Gyre Termes (formerly: qfonts/QuasiTimes)

```

The step package.

```

5007 \DeclareMicrotypeAlias{STEP-TLF}{ptmx}     % step
5008 \DeclareMicrotypeAlias{STEP-T0sF}{ptmj}     % "

```

The stix and stix2 packages (the latter has departed a bit from being a Times clone, but still seems close enough).

```

5009 \DeclareMicrotypeAlias{stix}{ptm}          % stix
5010 \DeclareMicrotypeAlias{stix2}{ptm}          % stix2

```

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

MicroPress's Charter version (chmath).

```
5011 %% -- Charter
5012 \DeclareMicrotypeAlias{chr}{bch} % CH Math
```

The XCharter package extends the Charter fonts.

```
5013 \DeclareMicrotypeAlias{XCharter-TLF}{bch} % XCharter
5014 \DeclareMicrotypeAlias{XCharter-T0sF}{bch} % "
```

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

```
5015 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
5016 %% -- Garamond
5017 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

The garamondx package, an extension of URW Garamond, providing small caps and oldstyle figures.

```
5018 \DeclareMicrotypeAlias{zgmX}{ugm} % garamondx
5019 \DeclareMicrotypeAlias{zgmj}{ugm} % "
5020 \DeclareMicrotypeAlias{zgmI}{ugm} % "
5021 \DeclareMicrotypeAlias{zgmQ}{ugm} % "
```

Because a configuration file for Adobe Garamond wouldn't be permitted for T_EX Live distribution, we use EB Garamond as the base font.

```
5022 \DeclareMicrotypeAlias{pad}{EBGaramond-LF}% Adobe Garamond
5023 \DeclareMicrotypeAlias{pdx}{EBGaramond-TLF}% "
5024 \DeclareMicrotypeAlias{padj}{EBGaramond-T0sF}% "
5025 %% --
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
5026 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

The eulervm package virtually extends the Euler fonts.

```
5027 \DeclareMicrotypeAlias{zeur}{eur} % Euler VM
5028 \DeclareMicrotypeAlias{zeus}{eus} % "
```

Euro symbol fonts, to save some files.

```
5029 \DeclareMicrotypeAlias{zpeus}{zpeu} % Adobe Euro sans -> serif
5030 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
5031
```

15.3 Interaction with babel

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

```
5032 %%% -----
5033 %%% INTERACTION WITH THE `babel' PACKAGE
5034
5035 \DeclareMicrotypeBabelHook
5036 {english,UKenglish,british,USenglish,american}
5037 {kerning=, spacing=nonfrench}
5038
5039 \DeclareMicrotypeBabelHook
5040 {french,français,acadian,canadien}
5041 {kerning=french, spacing=}
5042
5043 \DeclareMicrotypeBabelHook
5044 {turkish}
5045 {kerning=turkish, spacing=}
5046
```

15.4 Note on admissible characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```
\ : \textbackslash
{ : \textbraceleft
} : \textbraceright
^ : \textasciicircum
% : \%
# : \#
```

Comma and equal sign must be guarded with braces ('{,}', '{=}') to keep keyval happy.

Character commands are allowed as far as they have been defined in the proper \LaTeX way, that is, when they have been assigned a slot in the font encoding with $\text{\DeclareTextSymbol}$ or $\text{\DeclareTextComposite}$. Characters defined via \chardef are also possible.

Ligatures and \mathchardef ed symbols have to be specified numerically. Of course, numerical identification is possible in any other case, too.

8-bit characters are also admissible, provided they have been declared in the input encoding file. They should, however, only be used in private configuration files, where the proper input encoding is guaranteed, or else in combination with the 'inputenc' key.

With \XeTeX or \LuaTeX , in contrast, it is advisable to use the proper Unicode characters, or the font-specific glyph names prefixed with '/' (cf. section 16).

15.5 Character inheritance

First the lists of inheriting characters. We only declare those characters that are the same on *both* sides, i.e., not Œ for O.

```
5047 </m-t>
5048 <*m-t|zpeu|mys>
5049 %%% -----
5050 %%% CHARACTER INHERITANCE
5051
5052 </m-t|zpeu|mys>
5053 <*m-t>
```

15.5.1 OT1

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

```
5054 \DeclareCharacterInheritance
5055 { encoding = OT1 }
5056 { f = {011}, % ff
5057   i = {\i},
5058   j = {\j},
5059   0 = {\0},
5060   o = {\o}
5061 }
5062
```

15.5.2 T1

Candidates here: 028 (‘fi’), 029 (‘fl’), 030 (‘ffi’), 031 (‘ffl’), 156 (‘IJ’ ligature, since L^AT_EX 2005/12/01 accessible as \IJ), 188 (‘ij’, \ij), Æ, æ, Œ, œ.

```

5063 \DeclareCharacterInheritance
5064 { encoding = T1 }
5065 { A = {\^A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
5066   a = {\^a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
5067   C = {\'C,\c C,\v C},
5068   c = {\'c,\c c,\v c},
5069   D = {\v D,\DH},
5070   d = {\v d,\dj},
5071   E = {\^E,\'E,\^E,\"E,\k E,\v E},
5072   e = {\^e,\'e,\^e,\"e,\k e,\v e},
5073   f = {027}, % ff
5074   G = {\u G},
5075   g = {\u g},
5076   I = {\^I,\'I,\^I,\"I,\.I},
5077   i = {\^i,\'i,\^i,\"i,\i},
5078   j = {\j},
5079   L = {\L,\'L,\v L},
5080   l = {\l,\'l,\v l},
5081   N = {\'N,\~N,\v N},
5082   n = {\'n,\~n,\v n},
5083   O = {\0,\^0,\'0,\^0,\~0,\"0,\H 0},
5084   o = {\0,\^o,\'o,\^o,\~o,\"o,\H o},
5085   R = {\'R,\v R},
5086   r = {\'r,\v r},
5087   S = {\'S,\c S,\v S,\SS},
5088   s = {\'s,\c s,\v s},
5089   T = {\c T,\v T},
5090   t = {\c t,\v t},
5091   U = {\^U,\'U,\^U,\"U,\H U,\r U},
5092   u = {\^u,\'u,\^u,\"u,\H u,\r u},
5093   Y = {\'Y,\"Y},
5094   y = {\'y,\"y},
5095   Z = {\'Z,\.Z,\v Z},
5096   z = {\'z,\.z,\v z}

```

The ‘soft hyphen’ often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```

5097 % - = {127},
5098 }
5099

```

15.5.3 LY1

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

```

5100 \DeclareCharacterInheritance
5101 { encoding = LY1 }
5102 { A = {\^A,\'A,\^A,\~A,\"A,\r A},
5103   a = {\^a,\'a,\^a,\~a,\"a,\r a},
5104   C = {\c C},
5105   c = {\c c},
5106   D = {\DH},
5107   E = {\^E,\'E,\^E,\"E},
5108   e = {\^e,\'e,\^e,\"e},
5109   f = {011}, % ff
5110   I = {\^I,\'I,\^I,\"I},
5111   i = {\^i,\'i,\^i,\"i,\i},
5112   L = {\L},
5113   l = {\l},
5114   N = {\~N},

```

```

5115     n = {\~n},
5116     O = {\^0,\^0,\^0,\^0,\^0,\^0},
5117     o = {\^o,\^o,\^o,\^o,\^o,\^o},
5118     S = {\v S},
5119     s = {\v s},
5120     U = {\^U,\^U,\^U,\^U},
5121     u = {\^u,\^u,\^u,\^u},
5122     Y = {\^Y,\^Y},
5123     y = {\^y,\^y},
5124     Z = {\v Z},
5125     z = {\v z}
5126 }
5127

```

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.

```

5128 \DeclareCharacterInheritance
5129 { encoding = OT4 }
5130 { A = {\k A},
5131   a = {\k a},
5132   C = {\^C},
5133   c = {\^c},
5134   E = {\k E},
5135   e = {\k e},
5136   f = {011}, % ff
5137   i = {\i},
5138   j = {\j},
5139   L = {\L},
5140   l = {\l},
5141   N = {\^N},
5142   n = {\^n},
5143   O = {\0,\^0},
5144   o = {\o,\^o},
5145   S = {\^S},
5146   s = {\^s},
5147   Z = {\^Z,\^Z},
5148   z = {\^z,\^z},
5149   \textquotedblleft = "FF
5150 }
5151

```

15.5.5 QX

The Central European QX encoding.¹⁷ Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

5152 \DeclareCharacterInheritance
5153 { encoding = QX }
5154 { A = {\^A,\^A,\^A,\^A,\^A,\^A,\^A,\^A},
5155   a = {\^a,\^a,\^a,\^a,\^a,\^a,\^a,\^a},
5156   C = {\^C,\^C},
5157   c = {\^c,\^c},
5158   D = {\DH},
5159   E = {\^E,\^E,\^E,\^E,\^E,\^E},
5160   e = {\^e,\^e,\^e,\^e,\^e,\^e},
5161   f = {011}, % ff
5162   I = {\^I,\^I,\^I,\^I,\^I,\^I},
5163   i = {\^i,\^i,\^i,\^i,\^i,\^i},
5164   j = {\j},

```

17 Contributed by Maciej Eder.

```

5165 L = {\L},
5166 l = {\l},
5167 N = {\'N,\~N},
5168 n = {\'n,\~n},
5169 O = {\0,\`0,\'0,\^0,\-0,\"0},
5170 o = {\o,\`o,\'o,\^o,\-o,\"o},

```

The Romanian `\textcommabelow` accents are actually replacements for the `\c` variants, which had previously (and erroneously¹⁸) been included in QX encoding. They are still kept for backwards compatibility.

```

5171 S = {\'S,\c S,\textcommabelow S,\v S},
5172 s = {\'s,\c s,\textcommabelow s,\v s},
5173 T = {\c T,\textcommabelow T},
5174 t = {\c t,\textcommabelow t},
5175 U = {\`U,\'U,\^U,\"U,\k U},
5176 u = {\`u,\'u,\^u,\"u,\k u},
5177 Y = {\'Y,\"Y},
5178 y = {\'y,\"y},
5179 Z = {\'Z,\Z,\v Z},
5180 z = {\'z,\Z,\v z},
5181 . = \textellipsis
5182 }
5183

```

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.

```

5184 \DeclareCharacterInheritance
5185 { encoding = T5 }
5186 { A = {\`A,\'A,\-A,\h A,\d A,\^A,\u A,
5187       \`Acircumflex,\'Acircumflex,\-Acircumflex,\hAcircumflex,\dAcircumflex,
5188       \`Abreve,\'Abreve,\-Abreve,\hAbreve,\dAbreve},
5189   a = {\`a,\'a,\-a,\h a,\d a,\^a,\u a,
5190       \`acircumflex,\'acircumflex,\-acircumflex,\hacircumflex,\dacircumflex,
5191       \`abreve,\'abreve,\-abreve,\habreve,\dabreve},
5192   D = {\DJ},
5193   d = {\dj},
5194   E = {\`E,\'E,\-E,\h E,\d E,\^E,
5195       \`Ecircumflex,\'Ecircumflex,\-Ecircumflex,\hEcircumflex,\dEcircumflex},
5196   e = {\`e,\'e,\-e,\h e,\d e,\^e,
5197       \`ecircumflex,\'ecircumflex,\-ecircumflex,\hecircumflex,\decircumflex},
5198   I = {\`I,\'I,\-I,\h I,\d I},
5199   i = {\`i,\'i,\-i,\h i,\d i,\i},
5200   O = {\`O,\'O,\-O,\h O,\d O,\^O,\horn O,
5201       \`Ocircumflex,\'Ocircumflex,\-Ocircumflex,\hOcircumflex,\dOcircumflex,
5202       \`Ohorn,\'Ohorn,\-Ohorn,\hOhorn,\dOhorn},
5203   o = {\`o,\'o,\-o,\h o,\d o,\^o,\horn o,
5204       \`ocircumflex,\'ocircumflex,\-ocircumflex,\hocircumflex,\docircumflex,
5205       \`ohorn,\'ohorn,\-ohorn,\hohorn,\dohorn},
5206   U = {\`U,\'U,\-U,\h U,\d U,\horn U,
5207       \`Uhorn,\'Uhorn,\-Uhorn,\hUhorn,\dUhorn},
5208   u = {\`u,\'u,\-u,\h u,\d u,\horn u,
5209       \`uhorn,\'uhorn,\-uhorn,\huhorn,\duhorn},
5210   Y = {\`Y,\'Y,\-Y,\h Y,\d Y},
5211   y = {\`y,\'y,\-y,\h y,\d y}
5212 }
5213

```

18 Cf. <http://tug.org/pipermail/tex-live/2008-August/017204.html>

15.5.7 EU1, EU2, TU

The EU1 (X_YTeX), EU2 (LuaTeX), and, since fontspec version 2.5, TU encodings are not well-defined in the sense that they don't contain a fixed number of glyphs, all of which must be present. OpenType fonts may contain thousands of glyphs, but we only define those that should be present in every font (basically T1). This inheritance list should be overridden by font-specific ones.

```

5214 \DeclareCharacterInheritance
5215 { encoding = {TU,EU1,EU2} }
5216 { A = {\`A,\'A,\^A,\-A,\"A,\r A,\k A,\u A},
5217   a = {\`a,\'a,\^a,\-a,\"a,\r a,\k a,\u a},
5218   C = {\'C,\c C,\v C},
5219   c = {\'c,\c c,\v c},
5220   D = {\v D,\DH},
5221   d = {\v d,\dj},
5222   E = {\`E,\'E,\^E,\"E,\k E,\v E},
5223   e = {\`e,\'e,\^e,\"e,\k e,\v e},
5224 %   f = {/f_f}, % sometimes /f_f, sometimes /ff
5225   G = {\u G},
5226   g = {\u g},
5227   I = {\`I,\'I,\^I,\"I,\.I},
5228   i = {\`i,\'i,\^i,\"i,\i},
5229 %   j = {\j},
5230   L = {\L,\'L,\v L},
5231   l = {\l,\'l,\v l},
5232   N = {\'N,\-N,\v N},
5233   n = {\'n,\-n,\v n},
5234   O = {\0,\`0,\'0,\^0,\-0,\"0,\H 0},
5235   o = {\0,\`o,\'o,\^o,\-o,\"o,\H o},
5236   R = {\'R,\v R},
5237   r = {\'r,\v r},
5238   S = {\'S,\c S,\v S}, % \SS
5239   s = {\'s,\c s,\v s},
5240   T = {\c T,\v T},
5241   t = {\c t,\v t},
5242   U = {\`U,\'U,\^U,\"U,\H U,\r U},
5243   u = {\`u,\'u,\^u,\"u,\H u,\r u},
5244   Y = {\'Y,\"Y},
5245   y = {\'y,\"y},
5246   Z = {\'Z,\.Z,\v Z},
5247   z = {\'z,\.z,\v z}
5248 }
5249
5250 </m-t>

```

15.5.8 Euro symbols

Make Euro symbols settings simpler.

```

5251 <*>peu>
5252 \DeclareCharacterInheritance
5253 { encoding = U,
5254   family   = {zpeu,zpeus,eurosans} }
5255 { E = 128 }
5256
5257 </zpeu>
5258 <mv>s>

```

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.

```

5259 \DeclareCharacterInheritance
5260 { encoding = {OT1,U},

```

```

5261     family    = mvs }
5262 { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
5263
5264 </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.

```

5265 < *m-t >
5266 %%% -----
5267 %%% TRACKING/LETTERSPACING
5268
5269 \SetTracking
5270 [ name          = default,
5271   no ligatures = {f} ]
5272 { encoding      = {OT1,T1,T2A,LY1,OT4,QX,EU2,TU} }
5273 { }
5274

```

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).

```

5275 %%% -----
5276 %%% EXPANSION
5277
5278 \SetExpansion
5279 [ name          = default ]
5280 { encoding = {OT1,OT4,QX,T1,LY1} }
5281 {
5282   A = 500,      a = 700,
5283   \AE = 500,    \ae = 700,
5284   B = 700,      b = 700,
5285   C = 700,      c = 700,
5286   D = 500,      d = 700,
5287   E = 700,      e = 700,
5288   F = 700,
5289   G = 500,      g = 700,
5290   H = 700,      h = 700,
5291   K = 700,      k = 700,
5292   M = 700,      m = 700,
5293   N = 700,      n = 700,
5294   O = 500,      o = 700,
5295   \OE = 500,    \oe = 700,
5296   P = 700,      p = 700,
5297   Q = 500,      q = 700,
5298   R = 700,
5299   S = 700,      s = 700,
5300   U = 700,      u = 700,
5301   W = 700,      w = 700,
5302   Z = 700,      z = 700,
5303   2 = 700,
5304   3 = 700,
5305   6 = 700,
5306   8 = 700,
5307   9 = 700
5308 }
5309

```


Settings for Cyrillic T2A encoding.¹⁹

```

5310 \SetExpansion
5311   [ name      = T2A ]
5312   { encoding = T2A }
5313   {
5314     A = 500,      a = 700,
5315     B = 700,      b = 700,
5316     C = 700,      c = 700,
5317     D = 500,      d = 700,
5318     E = 700,      e = 700,
5319     F = 700,
5320     G = 500,      g = 700,
5321     H = 700,      h = 700,
5322     K = 700,      k = 700,
5323     M = 700,      m = 700,
5324     N = 700,      n = 700,
5325     O = 500,      o = 700,
5326     P = 700,      p = 700,
5327     Q = 500,      q = 700,
5328     R = 700,
5329     S = 700,      s = 700,
5330     U = 700,      u = 700,
5331     W = 700,      w = 700,
5332     Z = 700,      z = 700,
5333     2 = 700,
5334     3 = 700,
5335     6 = 700,
5336     8 = 700,
5337     9 = 700,
5338     \CYRA = 500,   \cyra = 700,
5339     \CYRB = 700,   \cyrb = 700,
5340     \CYRV = 700,   \cyrv = 700,
5341     \CYRG = 700,   \cyrg = 700,
5342     \CYRD = 700,   \cyrd = 700,
5343     \CYRE = 700,   \cyre = 700,
5344     \CYRZH = 700,  \cyrzh = 700,
5345     \CYRZ = 700,   \cyrz = 700,
5346     \CYRI = 700,   \cyri = 700,
5347     \CYRISHRT = 700, \cyrishrt = 700,
5348     \CYRK = 700,   \cyrk = 700,
5349     \CYRL = 700,   \cyr l = 700,
5350     \CYRM = 700,   \cyrm = 700,
5351     \CYRN = 700,   \cyrn = 700,
5352     \CYRO = 500,   \cyro = 700,
5353     \CYRP = 700,   \cyrp = 700,
5354     \CYRR = 700,   \cyrr = 700,
5355     \CYRS = 700,   \cyrs = 700,
5356     \CYRT = 700,   \cyrt = 700,
5357     \CYRU = 700,   \cyru = 700,
5358     \CYRF = 700,   \cyrf = 700,
5359     \CYRH = 700,   \cyrh = 700,
5360     \CYRC = 700,   \cyrc = 700,
5361     \CYRCH = 700,  \cyrch = 700,
5362     \CYRSH = 700,  \cyrsh = 700,
5363     \CYRSHCH = 700, \cyrshch = 700,
5364     \CYRHRSN = 700, \cyrhdsn = 700,
5365     \CYRERY = 700, \cyrery = 700,
5366     \CYRSFTSN = 700, \cyrstsn = 700,
5367     \CYREREV = 700, \cyrerev = 700,
5368     \CYRYU = 700,  \cyryu = 700,
5369     \CYRYA = 700,  \cyrya = 700
5370   }
5371

```

19 Contributed by *Karl Karlsson*.

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

```

5372 \SetExpansion
5373   [ name      = T5 ]
5374   { encoding = T5 }
5375   {
5376     A = 500,      a = 700,
5377     B = 700,      b = 700,
5378     C = 700,      c = 700,
5379     D = 500,      d = 700,
5380     E = 700,      e = 700,
5381     F = 700,
5382     G = 500,      g = 700,
5383     H = 700,      h = 700,
5384     K = 700,      k = 700,
5385     M = 700,      m = 700,
5386     N = 700,      n = 700,
5387     O = 500,      o = 700,
5388     P = 700,      p = 700,
5389     Q = 500,      q = 700,
5390     R = 700,
5391     S = 700,      s = 700,
5392     U = 700,      u = 700,
5393     W = 700,      w = 700,
5394     Z = 700,      z = 700,
5395     2 = 700,
5396     3 = 700,
5397     6 = 700,
5398     8 = 700,
5399     9 = 700
5400   }
5401
5402 /m-t

```

15.8 Character protrusion

```

5403 %%% -----
5404 %%% PROTRUSION
5405

```

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

```

\SetProtrusion
[ name      = thanh ]
{ encoding = OT1 }
{
  A = {50,50},
  F = { ,50},
  J = {50, },
  K = { ,50},
  L = { ,50},
  T = {50,50},
  V = {50,50},
  W = {50,50},
  X = {50,50},
  Y = {50,50},
  k = { ,50},
  r = { ,50},
  t = { ,50},
  v = {50,50},
  w = {50,50},
  x = {50,50},
  y = {50,50},
  . = { ,700},    {,}= { ,700},

```

```

: = { ,500},      ; = { ,500},
! = { ,200},      ? = { ,200},
( = {50, },      ) = { ,50},
- = { ,700},
\textendash      = { ,300},      \textemdash      = { ,200},
\textquoteleft   = {700, },      \textquoteright  = { ,700},
\textquotedblleft = {500, },      \textquotedblright = { ,500}
}

```

15.8.1 Normal

The default settings always use the most moderate value.

```

5406 <*cfg-t>
5407 \SetProtrusion
5408 <m-t> [ name = default ]

```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```

5409 <bch> [ name = bch-default ]

```

- Bitstream Letter Gothic (blg)

```

5410 <blg> [ name = blg-default ]

```

- Computer Modern Roman (cmr)

```

5411 <cmr> [ name = cmr-default ]

```

- EB Garamond

```

5412 <ebg> [ name = EBGaramond-default ]

```

- Minion²⁰ (pmnx, pmnj)

```

5413 <pmn> [ name = pmnj-default ]

```

- Palatino (ppl, pplx, pplj)

```

5414 <ppl> [ name = ppl-default ]

```

- Times (ptm, ptmx, ptmj)

```

5415 <ptm> [ name = ptm-default ]

```

- URW Garamond (ugm)

```

5416 <ugm> [ name = ugm-default ]
5417 <m-t|cmr|pmn> { }
5418 <bch|blg|ebg|ugm> { encoding = OT1,
5419 <ppl|ptm> { encoding = {OT1,OT4},
5420 <bch> family = bch }
5421 <blg> family = blg }
5422 <ebg> family = {EBGaramond-LF,EBGaramond-Of,EBGaramond-TLF,EBGaramond-TOf } }
5423 <ppl> family = {ppl,pplx,pplj } }
5424 <ptm> family = {ptm,ptmx,ptmj } }
5425 <ugm> family = ugm }
5426 {
5427 <m-t|bch|blg|cmr|ebg|pmn|ppl|ptm> A = {50,50},
5428 <ugm> A = {50,100},
5429 <ebg|ptm> \AE = {50, },
5430 <ugm> \AE = {150,50},
5431 <ugm> B = { ,50},
5432 <bch|ebg|pmn|ugm> C = {50, },

```

20 Contributed by Harald Harders and Karl Karlsson.

```

5433 <bch|ebg|pmn>      D = { ,50},
5434 <ugm>               D = { ,70},
5435 <ugm>               E = { ,50},
5436 <m-t|bch|cmr|ebg|pmn|ptm>      F = { ,50},
5437 <ugm>               F = { ,70},
5438 <bch|ebg|pmn>      G = {50, },
5439 <ugm>               G = {50,50},
5440 <blg>               I = {150,150},
5441 <m-t|cmr|ebg|pmn|ppl|ptm|ugm>      J = {50, },
5442 <bch|blg>          J = {100, },
5443 <!blg>              K = { ,50},
5444 <blg>               K = {50, },
5445 <m-t|bch|cmr|ebg|pmn|ppl>      L = { ,50},
5446 <blg>               L = { ,150},
5447 <ptm>               L = { ,80},
5448 <ugm>               L = { ,120},
5449 <bch|ebg|pmn|ugm>      O = {50,50},
5450 <ebg>               \OE = {50, },
5451 <ugm>               \OE = {50,50},
5452 <blg>               P = { ,100},
5453 <ugm>               P = { ,50},
5454 <bch|ebg|pmn>      Q = {50,70},
5455 <ugm>               Q = {50,50},
5456 <bch>               R = { ,50},
5457 <ugm>               R = { ,70},
5458 <m-t|bch|cmr|ebg|pmn|ppl|ptm>      T = {50,50},
5459 <blg>               T = {100,100},
5460 <ugm>               T = {70,70},
5461 <m-t|bch|cmr|ebg|pmn|ppl|ptm>      V = {50,50},
5462 <blg|ugm>          V = {70,70},
5463 <m-t|bch|cmr|ebg|pmn|ppl|ptm>      W = {50,50},
5464 <ugm>               W = {70,70},
5465 <m-t|bch|cmr|ebg|pmn|ppl|ptm>      X = {50,50},
5466 <ugm>               X = {50,70},
5467 <m-t|bch|cmr|ebg|pmn|ppl>      Y = {50,50},
5468 <blg|ptm|ugm>      Y = {80,80},
5469 <ugm>               Z = {50,50},
5470 <blg>               f = {150,100},
5471 <blg>               i = {150,150},
5472 <blg>               j = {100,100},
5473 <m-t|bch|cmr|ebg|pmn|ppl|ptm>      k = { ,50},
5474 <ugm>               k = { ,70},
5475 <blg>               l = {150,150},
5476 <pmn>               l = { , -50},
5477 <ebg|ppl>          p = {50,50},
5478 <ugm>               p = { ,50},
5479 <ebg|ppl>          q = {50, },
5480 <!blg>              r = { ,50},
5481 <blg>               r = {100, 80},
5482 <cmr|ebg|pmn>      t = { ,70},
5483 <bch>               t = { ,50},
5484 <blg>               t = {150, 80},
5485 <ugm>               t = { ,100},
5486 <m-t|bch|cmr|ebg|pmn|ppl|ptm>      v = {50,50},
5487 <blg>               v = {100,100},
5488 <ugm>               v = {50,70},
5489 <m-t|bch|cmr|ebg|pmn|ppl|ptm>      w = {50,50},
5490 <ugm>               w = {50,70},
5491 <!blg>              x = {50,50},
5492 <blg>               x = {100,100},
5493 <m-t|bch|ebg|pmn>      y = { ,50},
5494 <blg>               y = { 50,100},
5495 <cmr|ppl|ptm>      y = {50,70},
5496 <ugm>               y = { ,70},
5497 <cmr>              O = { ,50},

```

```

5498 <m-t>      1 = {50,50},
5499 <bch|blg|ebg|ptm|ugm>      1 = {150,150},
5500 <cmr>      1 = {100,200},
5501 <pmn>      1 = { ,50},
5502 <ppl>      1 = {100,100},
5503 <bch|cmr|ebg|ugm>      2 = {50,50},
5504 <blg>      2 = { ,100},
5505 <bch|pmn>      3 = {50, },
5506 <cmr|ebg|ugm>      3 = {50,50},
5507 <blg>      3 = {100, },
5508 <m-t|ebg>      4 = {50,50},
5509 <bch>      4 = {100,50},
5510 <blg>      4 = {100, },
5511 <cmr|ugm>      4 = {70,70},
5512 <pmn>      4 = {50, },
5513 <ptm>      4 = {70, },
5514 <cmr>      5 = { ,50},
5515 <ebg>      5 = {50,50},
5516 <bch>      6 = {50, },
5517 <cmr>      6 = { ,50},
5518 <ebg>      6 = {50,50},
5519 <m-t>      7 = {50,50},
5520 <bch|ebg|pmn|ugm>      7 = {50,80},
5521 <blg>      7 = {100,100},
5522 <cmr|ptm>      7 = {50,100},
5523 <ppl>      7 = { ,50},
5524 <cmr>      8 = { ,50},
5525 <bch|ebg>      9 = {50,50},
5526 <cmr>      9 = { ,50},
5527 <m-t|cmr|ebg|pmn|ppl|ptm|ugm>      . = { ,700},
5528 <bch>      . = { ,600},
5529 <blg>      . = {400,500},
5530 <!blg>      {,}= { ,500},
5531 <blg>      {,}= {300,400},
5532 <m-t|cmr|ebg|pmn|ppl|ptm|ugm>      : = { ,500},
5533 <bch>      : = { ,400},
5534 <blg>      : = {300,400},
5535 <m-t|bch|ebg|pmn|ptm>      ; = { ,300},
5536 <blg>      ; = {200,300},
5537 <cmr|ppl>      ; = { ,500},
5538 <ugm>      ; = { ,400},
5539 <!blg>      ! = { ,100},
5540 <blg>      ! = {200,200},
5541 <m-t|ebg|pmn|ptm>      ? = { ,100},
5542 <bch|cmr|ppl|ugm>      ? = { ,200},
5543 <blg>      ? = {150,150},
5544 <pmn>      " = {300,300},
5545 <m-t|bch|cmr|ebg|pmn|ppl>      @ = {50,50},
5546 <ptm>      @ = {100,100},
5547 <m-t|bch|blg|cmr|ebg|pmn|ppl|ptm>      ~ = {200,250},
5548 <ugm>      ~ = {300,350},
5549 <ebg|ppl|ptm>      & = {50,100},
5550 <ugm>      & = { ,100},
5551 <m-t|cmr|ebg|pmn>      \% = {50,50},
5552 <bch>      \% = { ,50},
5553 <ppl|ptm>      \% = {100,100},
5554 <ugm>      \% = {50,100},
5555 <blg>      \# = {100,100},
5556 <m-t|ppl|ptm|ugm>      * = {200,200},
5557 <bch|pmn>      * = {200,300},
5558 <blg>      * = {150,200},
5559 <cmr|ebg>      * = {300,300},
5560 <m-t|cmr|ppl|ptm>      + = {250,250},
5561 <bch>      + = {150,250},
5562 <ebg>      + = {300,300},

```

```

5563 <blg|pmn>      + = {150,200},
5564 <ugm>          + = {250,300},
5565 <blg|ugm>       {=} = {200,200},
5566 <m-t|ebg|pmn|ptm> ( = {100,   },   ) = {   ,200},
5567 <bch|ugm>        ( = {200,   },   ) = {   ,200},
5568 <cmr|blg>        ( = {300,   },   ) = {   ,300},
5569 <ppl>            ( = {100,   },   ) = {   ,300},
5570 <bch|pmn>        [ = {100,   },   ] = {   ,100},
5571 <blg>            [ = {300,100},   ] = {   ,300},

5572 <m-t|ebg|pmn|ptm> / = {100,200},
5573 <bch>            / = {   ,200},
5574 <blg>            / = {300,300},
5575 <cmr|ppl>        / = {200,300},
5576 <ugm>            / = {100,300},
5577 <m-t|ptm>        - = {500,500},
5578 <bch|cmr|ppl>    - = {400,500},
5579 <blg>            - = {300,400},
5580 <ebg>            - = {300,500},
5581 <pmn>            - = {200,400},
5582 <ugm>            - = {500,600},
5583 <blg>            < = {200,100},   > = {100,200},
5584 <blg>            _ = {150,250},
5585 <blg>            | = {250,250},
5586 <m-t|pmn>        \textendash      = {200,200},   \textendash      = {150,150},
5587 <bch>            \textendash      = {200,300},   \textendash      = {150,250},
5588 <cmr>            \textendash      = {400,300},   \textendash      = {300,200},
5589 <ebg|ppl|ptm>    \textendash      = {300,300},   \textendash      = {200,200},
5590 <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).

```

5591 <m-t|bch|pmn>    \textquoteleft    = {300,400},   \textquoteright    = {300,400},
5592 <blg>            \textquoteleft    = {400,600},   \textquoteright    = {400,600},
5593 <cmr>            \textquoteleft    = {500,700},   \textquoteright    = {500,600},
5594 <ebg|ppl>        \textquoteleft    = {500,700},   \textquoteright    = {500,700},
5595 <ptm>            \textquoteleft    = {500,500},   \textquoteright    = {300,500},
5596 <ugm>            \textquoteleft    = {300,600},   \textquoteright    = {300,600},
5597 <m-t|bch|pmn>    \textquotedblleft = {300,300},   \textquotedblright = {300,300}
5598 <blg>            \textquotedblright = {300,400}
5599 <cmr>            \textquotedblleft = {500,300},   \textquotedblright = {200,600}
5600 <ebg|ppl|ptm>    \textquotedblleft = {300,400},   \textquotedblright = {300,400}
5601 <ugm>            \textquotedblleft = {400,400},   \textquotedblright = {400,400}
5602 }
5603

```

Greek uppercase letters are in OT1 encoding only.

```

5604 <*m-t|cmr|pmn>
5605 \SetProtrusion
5606 <m-t> [ name      = OT1-default,
5607 <cmr> [ name      = cmr-OT1,
5608 <pmn> [ name      = pmnj-OT1,
5609 <m-t> load        = default ]
5610 <cmr> load        = cmr-default ]
5611 <pmn> load        = pmnj-default ]
5612 <m-t> { encoding = OT1 }
5613 <cmr> { encoding = {OT1,OT4},
5614 <pmn> { encoding = OT1,
5615 <cmr> family    = cmr }
5616 <pmn> family    = pmnj }
5617 {
5618 <m-t|cmr> \AE = {50,   },
5619 <pmn>     \OE = {50,   }
5620 <*cmr>
5621 "00 = {   ,150}, % \Gamma

```

```

5622 "01 = {100,100}, % \Delta
5623 "02 = { 50, 50}, % \Theta
5624 "03 = {100,100}, % \Lambda
5625 "06 = { 50, 50}, % \Sigma
5626 "07 = {100,100}, % \Upsilon
5627 "08 = { 50, 50}, % \Phi
5628 "09 = { 50, 50} % \Psi

```

Remaining slots can be found in the source file.

```

5629 </cmr>
5630 }
5631
5632 </m-t|cmr|pmn>

```

T1 and LY1 encodings contain some more characters. The default list will be loaded first. For X_YTeX (EU1) and LuaTeX (EU2) we simply use the T1 list as default (for now).

```

5633 \SetProtrusion
5634 <m-t> [ name = T1-default,
5635 <bch> [ name = bch-T1,
5636 <blg> [ name = blg-T1,
5637 <cmr> [ name = cmr-T1,
5638 <ebg> [ name = EBGaramond-T1,
5639 <pmn> [ name = pmnj-T1,
5640 <ppl> [ name = ppl-T1,
5641 <ptm> [ name = ptm-T1,
5642 <ugm> [ name = ugm-T1,
5643 <m-t> load = default ]
5644 <bch> load = bch-default ]
5645 <blg> load = blg-default ]
5646 <cmr> load = cmr-default ]
5647 <ebg> load = EBGaramond-default ]
5648 <pmn> load = pmnj-default ]
5649 <ppl> load = ppl-default ]
5650 <ptm> load = ptm-default ]
5651 <ugm> load = ugm-default ]
5652 <m-t> { encoding = {T1,LY1,EU1,EU2,TU} }
5653 <bch|cmr|ebg|pmn|ppl> { encoding = {T1,LY1},
5654 <blg|ptm|ugm> { encoding = {T1},
5655 <bch> family = bch }
5656 <blg> family = blg }
5657 <cmr> family = cmr }
5658 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-Of,EBGaramond-TOf} }
5659 <pmn> family = pmnj }
5660 <ppl> family = {ppl,pplx,pplj} }
5661 <ptm> family = {ptm,ptmx,ptmj} }
5662 <ugm> family = ugm }
5663 {
5664 <m-t|cmr> \AE = {50, },
5665 <bch|pmn> \OE = {50, },
5666 <pmn> \TH = { ,50},
5667 <blg> \v L = { ,250},
5668 <blg> \v d = { ,250},
5669 <blg> \v l = { ,250},
5670 <blg> \v t = { ,250},
5671 <blg> 127 = {300,400},
5672 <blg> 156 = {100, }, % IJ
5673 <blg> 188 = { 80, 80}, % ij
5674 <m-t|bch|ebg|pmn|ppl|ptm> _ = {100,100},
5675 <cmr> _ = {200,200},
5676 <ugm> _ = {100,200},
5677 <m-t|ebg|pmn|ptm> \textbackslash = {100,200},
5678 <bch> \textbackslash = {150,200},
5679 <blg> \textbackslash = {250,300},
5680 <cmr|ppl> \textbackslash = {200,300},

```

```

5681 <ugm> \textbackslash = {100,300},
5682 <ugm> \textbar = {200,200},
5683 <blg> \textendash = {300,300}, \textemdash = {150,150},
5684 <blg> \textquotedbl = {300,400}, \textquotedblleft = {300,400},
5685 <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.

```

5686 <m-t|cmr|ebg|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
5687 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
5688 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
5689 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
5690 <blg> \guilsinglleft = {300,500}, \guilsinglright = {300,500},
5691 <cmr|ebg|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5692 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
5693 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5694 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5695 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
5696 <blg|ebg|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
5697 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
5698 <m-t|bch|cmr|ebg|pmn|ppl|ugm> \texttexclamdown = {100, }, \textquestiondown = {100, },
5699 <blg> \texttexclamdown = {200, }, \textquestiondown = {100, },
5700 <ptm> \texttexclamdown = {200, }, \textquestiondown = {200, },
5701 <m-t|cmr|ebg|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
5702 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
5703 <m-t|bch|cmr|ebg|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200},
5704 <pmn> \textless = {100, }, \textgreater = { ,100},
5705 <pmn> \textvisiblespace = {100,100} % not in LY1
5706 }
5707

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```

5708 <*cmr>
5709 \SetProtrusion
5710 [ name = lmr-T1,
5711   load = cmr-T1 ]
5712 { encoding = {T1,LY1},
5713   family = lmr }
5714 {
5715   \textquotedblleft = {300,400}, \textquotedblright = {300,400}
5716 }
5717
5718 </cmr>

```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).²¹

```

5719 <*m-t|cmr|pmn>
5720 \SetProtrusion
5721 <m-t> [ name = T2A-default,
5722 <cmr> [ name = cmr-T2A,
5723 <pmn> [ name = pmnj-T2A,
5724 <m-t> load = default ]
5725 <cmr> load = cmr-default ]
5726 <pmn> load = pmnj-default ]
5727 { encoding = T2A,
5728 <m-t> }
5729 <cmr> family = cmr }
5730 <pmn> family = pmnj }
5731 {
5732   \CYRA = {50,50},
5733   \CYRG = { ,50},
5734   \CYRK = { ,50},

```

21 Contributed by Karl Karlsson.


```

5735 \CYRT = {50,50},
5736 \CYRH = {50,50},
5737 \CYRU = {50,50},
5738 <pmn> \CYRS = {50, },
5739 <pmn> \CYRO = {50,50},
5740 \cyrk = { ,50},
5741 \cyrp = { ,50},
5742 \cyrh = {50,50},
5743 <m-t|pmn> \cyru = {50,50},
5744 <cmr> \cyru = {50,70},
5745 <m-t> _ = {100,100},
5746 <cmr> _ = {200,200},
5747 <m-t> \textbackslash = {100,200}, \quotedblbase = {400,400},
5748 <cmr> \textbackslash = {200,300}, \quotedblbase = {400,400},
5749 <pmn> \textbackslash = {100,200}, \quotedblbase = {300,300},
5750 <cmr> \textquotedbl = {300,300}, \textquotedblleft = {200,600},
5751 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5752 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5753 <pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
5754 <m-t|cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
5755 <pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
5756 <m-t|cmr> \textless = {200,100}, \textgreater = {100,200}
5757 <pmn> \textless = {100, }, \textgreater = { ,100}
5758 }
5759
5760 </m-t|cmr|pmn>

```

Settings for the QX encoding (generic and Times).²² It also includes some glyphs otherwise in TS1.

```

5761 <*m-t|ptm>
5762 \SetProtrusion
5763 <m-t> [ name = QX-default,
5764 <ptm> [ name = ptm-QX,
5765 <m-t> load = default ]
5766 <ptm> load = ptm-default ]
5767 <m-t> { encoding = QX }
5768 <ptm> { encoding = QX,
5769 <ptm> family = {ptm,ptmx,ptmj} }
5770 {
5771 \AE = {50, },
5772 <ptm> * = {200,200},
5773 {} = {100,100},
5774 \textunderscore = {100,100},
5775 \textbackslash = {100,200},
5776 \quotedblbase = {400,400},
5777 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5778 <ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
5779 \textexclamdown = {100, }, \textquestiondown = {100, },
5780 <m-t> \textbraceleft = {400,200}, \textbraceright = {200,400},
5781 <ptm> \textbraceleft = {200,200}, \textbraceright = {200,300},
5782 \textless = {200,100}, \textgreater = {100,200},
5783 \textminus = {200,200}, \textdegree = {300,300},
5784 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5785 <ptm> \copyright = {100,150}, \textregistered = {100,150},
5786 <ptm> \textxgeq = { ,100}, \textxleq = {100, },
5787 <ptm> \textalpha = { , 50}, \textDelta = { 70, 70},
5788 <ptm> \textpi = { 50, 80}, \textSigma = { , 70},
5789 <ptm> \textmu = { , 80}, \texteuro = { 50, 50},
5790 <ptm> \textellipsis = {150,200}, \textasciitilde = { 80, 80},
5791 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
5792 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5793 <ptm> \textdiv = { 50,150}, \textsection = { 80, 80},
5794 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},

```

22 Contributed by Maciej Eder.

```

5795 <ptm> \textbullet = {150,150}, \textperiodcentered = {300,300},
5796 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5797 <ptm> \textperthousand = { ,50}
5798 }
5799
5800 </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.

```

5801 <*cmr|bch>
5802 \SetProtrusion
5803 <cmr> [ name = cmr-T5,
5804 <cmr> load = cmr-default ]
5805 <bch> [ name = bch-T5,
5806 <bch> load = bch-default ]
5807 { encoding = T5,
5808 <cmr> family = cmr }
5809 <bch> family = bch }
5810 {
5811 <bch> _ = {100,100},
5812 <bch> \textbackslash = {150,200},
5813 <cmr> \textbackslash = {200,300},
5814 <cmr> \textquotedblleft = {200,600},
5815 <cmr> \textquotedbl = {300,300},
5816 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
5817 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
5818 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
5819 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5820 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},
5821 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5822 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
5823 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
5824 \textless = {200,100}, \textgreater = {100,200}
5825 }
5826
5827 </cmr|bch>

```

Minion with lining numbers.

```

5828 <*pmn>
5829 \SetProtrusion
5830 [ name = pmnx-OT1,
5831 load = pmnj-default ]
5832 { encoding = OT1,
5833 family = pmnx }
5834 {
5835 1 = {230,180}
5836 }
5837
5838 \SetProtrusion
5839 [ name = pmnx-T1,
5840 load = pmnj-T1 ]
5841 { encoding = {T1,LY1},
5842 family = pmnx }
5843 {
5844 1 = {230,180}
5845 }
5846
5847 \SetProtrusion
5848 [ name = pmnx-T2A,
5849 load = pmnj-T2A ]
5850 { encoding = {T2A},
5851 family = pmnx }
5852 {
5853 1 = {230,180}
5854 }

```

```
5855
5856 </pmn>
```

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

```
5857 <*ptm>
5858 \SetProtrusion
5859 [ name      = ptm-LY1,
5860   load      = ptm-T1 ]
5861 { encoding = LY1,
5862   family   = {ptm,ptmx,ptmj} }
5863 {
5864   _                = {100,100},
5865   \texttrademark   = {100,100},
5866   \textregistered  = {100,100},
5867   \textcopyright   = {100,100},
5868   \textdegree      = {300,300},
5869   \textminus       = {200,200},
5870   \textellipsis    = {150,200},
5871   \% \texteuro     = { , }, % ?
5872   \textcent        = {100,100},
5873   \textquotesingle = {500,500},
5874   \textflorin      = { 50, 70},
5875   \textdagger       = {150,150},
5876   \textdaggerdbl    = {100,100},
5877   \textperthousand = { , 50},
5878   \textbullet       = {150,150},
5879   \textonesuperior = {100,100},
5880   \texttwosuperior  = { 50, 50},
5881   \textthreesuperior = { 50, 50},
5882   \textperiodcentered = {300,300},
5883   \textplusminus    = { 50, 80},
5884   \textmultiply     = {100,100},
5885   \textdivide       = { 50,150}
```

Remaining slots in the source file.

```
5886   }
5887
5888 </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. In the generic settings we therefore omit the letters, and only set up the punctuation characters.

The italic glyphs of Computer Modern Roman feature a lot of side bearing, therefore almost all of them have to protrude.²³

```
5889 \SetProtrusion
5890 <m-t> [ name      = OT1-it  ]
5891 <bch> [ name      = bch-it  ]
5892 <blg> [ name      = blg-it,
5893 <blg>   load      = blg-default ]
5894 <cmr> [ name      = cmr-it  ]
5895 <ebg> [ name      = EBGaramond-it ]
5896 <pmn> [ name      = pmnj-it ]
5897 <ppl> [ name      = ppl-it  ]
5898 <ptm> [ name      = ptm-it  ]
5899 <ugm> [ name      = ugm-it  ]
5900 <m-t|bch|blg|ebg|ugm> { encoding = OT1,
```

23 Settings contributed by Hendrik Vogt.

```

5901 < ppl | ptm > { encoding = {OT1,OT4},
5902 < bch > family = bch,
5903 < blg > family = blg,
5904 < ebg > family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OfF,EBGaramond-TOfF},
5905 < ppl > family = {ppl,pplx,pplj},
5906 < ptm > family = {ptm,ptmx,ptmj},
5907 < ugm > family = ugm,
5908 < m-t | bch | ebg | ppl | ptm > shape = {it,sl} }
5909 < blg | ugm > shape = it }
5910 < cmr | pmn > { }
5911 {
5912 < cmr > A = {100,100},
5913 < ptm > A = {100,50},
5914 < ebg | pmn > A = {50, },
5915 < ugm > A = { ,150},
5916 < ppl > A = {50,50},
5917 < ptm > \AE = {100, },
5918 < ebg | ppl > \AE = {50, },
5919 < cmr > B = {83,-40},
5920 < ebg | ppl | ptm > B = {50, },
5921 < pmn > B = {20,-50},
5922 < bch | ppl | ptm | ugm > C = {50, },
5923 < cmr > C = {165,-75},
5924 < ebg > C = {100, },
5925 < pmn > C = {50,-50},
5926 < cmr > D = {75, -28},
5927 < ebg | ppl | ptm > D = {50,50},
5928 < pmn > D = {20, },
5929 < cmr > E = {80,-55},
5930 < ebg | ppl | ptm > E = {50, },
5931 < pmn > E = {20,-50},
5932 < cmr > F = {85,-80},
5933 < ebg | ptm > F = {100, },
5934 < pmn > F = {10, },
5935 < ppl > F = {50, },
5936 < bch | ppl | ptm | ugm > G = {50, },
5937 < cmr > G = {153,-15},
5938 < ebg > G = {100, },
5939 < pmn > G = {50,-50},
5940 < cmr > H = {73,-60},
5941 < ebg | ppl | ptm > H = {50, },
5942 < cmr > I = {140,-120},
5943 < ebg | ptm > I = {50, },
5944 < pmn > I = {20,-50},
5945 < cmr > J = {135,-80},
5946 < ebg > J = {50, },
5947 < pmn > J = {20, },
5948 < ptm > J = {100, },
5949 < cmr > K = {70,-30},
5950 < ebg | ppl | ptm > K = {50, },
5951 < pmn > K = {20, },
5952 < cmr > L = {87, 40},
5953 < ebg | ppl | ptm > L = {50, },
5954 < pmn > L = {20,50},
5955 < ugm > L = { ,100},
5956 < cmr > M = {67,-45},
5957 < pmn > M = { , -30},
5958 < ptm > M = {50, },
5959 < cmr > N = {75,-55},
5960 < pmn > N = { , -30},
5961 < ptm > N = {50, },
5962 < bch | pmn | ppl | ptm > O = {50, },
5963 < cmr > O = {150,-30},
5964 < ebg > O = {100, },
5965 < ugm > O = {70,50},

```

```

5966 <ppl|ptm> \OE = {50, },
5967 <ebg> \OE = {100, },
5968 <cmr> P = {82,-50},
5969 <ebg|ppl|ptm> P = {50, },
5970 <pmn> P = {20,-50},
5971 <bch|pmn|ppl|ptm> Q = {50, },
5972 <cmr> Q = {150,-30},
5973 <ebg> Q = {100, },
5974 <ugm> Q = {70,50},
5975 <cmr> R = {75, 15},
5976 <ebg|ppl|ptm> R = {50, },
5977 <pmn> R = {20, },
5978 <bch|ebg|ppl|ptm> S = {50, },
5979 <cmr> S = {90,-65},
5980 <pmn> S = {20,-30},
5981 <bch|ebg|ppl|ptm> $ = {50, },
5982 <cmr> $ = {100,-20},
5983 <pmn> $ = {20,-30},
5984 <bch|pmn|ugm> T = {70, },
5985 <cmr> T = {220,-85},
5986 <ebg|ppl|ptm> T = {100, },
5987 <cmr> U = {230,-55},
5988 <ebg|ppl|ptm> U = {50, },
5989 <pmn> U = {50,-50},
5990 <cmr> V = {260,-60},
5991 <ebg|pmn|ugm> V = {100, },
5992 <ppl|ptm> V = {100,50},
5993 <cmr> W = {185,-55},
5994 <ebg|pmn|ugm> W = {100, },
5995 <ppl> W = {50, },
5996 <ptm> W = {100,50},
5997 <cmr> X = {70,-30},
5998 <ppl|ptm> X = {50, },
5999 <cmr> Y = {250,-60},
6000 <pmn> Y = {50, },
6001 <ppl> Y = {100,50},
6002 <ptm> Y = {100, },
6003 <cmr> Z = {90,-60},
6004 <pmn> Z = { , -50},
6005 <cmr> a = {150,-10},
6006 <cmr> b = {170, },
6007 <cmr> c = {173,-10},
6008 <cmr> d = {150,-55},
6009 <pmn> d = { , -50},
6010 <cmr> e = {180, },
6011 <cmr> f = { , -250},
6012 <ebg|pmn> f = { , -100},
6013 <cmr> g = {150,-10},
6014 <cmr> h = {100, },
6015 <cmr> i = {210, },
6016 <pmn> i = { , -30},
6017 <cmr> j = { , -40},
6018 <pmn> j = { , -30},
6019 <cmr> k = {110,-50},
6020 <cmr> l = {240,-110},
6021 <pmn> l = { , -100},
6022 <cmr> m = {80, },
6023 <cmr> n = {115, },
6024 <bch> o = {50,50},
6025 <cmr> o = {155, },
6026 <bch> p = { , 50},
6027 <pmn> p = {-50, },
6028 <bch> q = {50, },
6029 <cmr> q = {170,-40},
6030 <cmr> r = {155,-40},

```

```

6031 <pmn>      r = { ,50},
6032 <cmr>      s = {130, },
6033 <bch>      t = { ,50},
6034 <cmr>      t = {230,-10},
6035 <cmr>      u = {120, },
6036 <cmr>      v = {140,-25},
6037 <pmn|ugm>  v = {50, },
6038 <bch>      w = { ,50},
6039 <cmr>      w = {98,-20},
6040 <pmn|ugm>  w = {50, },
6041 <cmr>      x = {65,-40},
6042 <bch>      y = { ,50},
6043 <cmr>      y = {130,-20},
6044 <cmr>      z = {110,-80},
6045 <cmr>      0 = {170,-85},
6046 <bch|ptm>  1 = {150,100},
6047 <cmr>      1 = {230,110},
6048 <ebg>      1 = {150, },
6049 <pmn>      1 = {50, },
6050 <ppl>      1 = {100, },
6051 <ugm>      1 = {150,150},
6052 <cmr>      2 = {130,-70},
6053 <ebg|ppl|ptm> 2 = {50, },
6054 <pmn>      2 = {-50, },
6055 <bch>      3 = {50, },
6056 <cmr>      3 = {140,-70},
6057 <pmn>      3 = {-100, },
6058 <ptm>      3 = {100,50},
6059 <bch>      4 = {100, },
6060 <cmr>      4 = {130,80},
6061 <ebg>      4 = {150, },
6062 <ppl|ptm> 4 = {50, },
6063 <cmr>      5 = {160, },
6064 <ptm>      5 = {50, },
6065 <bch>      6 = {50, },
6066 <cmr>      6 = {175,-30},
6067 <bch|ebg|ptm> 7 = {100, },
6068 <cmr>      7 = {250,-150},
6069 <pmn>      7 = {20, },
6070 <ppl>      7 = {50, },
6071 <cmr>      8 = {130,-40},
6072 <cmr>      9 = {155,-80},
6073 <m-t|cmr|ebg|pmn|ppl> . = { ,500},
6074 <blg>      . = {400,600},
6075 <bch|ptm|ugm> . = { ,700},
6076 <blg>      {,}= {300,500},
6077 <m-t|ebg|pmn|ppl> {,}= { ,500},
6078 <cmr>      {,}= { ,450},
6079 <bch|ugm>   {,}= { ,600},
6080 <ptm>      {,}= { ,700},
6081 <m-t|cmr|ebg|ppl> : = { ,300},
6082 <bch|ugm>   : = { ,400},
6083 <pmn>      : = { ,200},
6084 <ptm>      : = { ,500},
6085 <m-t|cmr|ebg|ppl> ; = { ,300},
6086 <bch|ugm>   ; = { ,400},
6087 <pmn>      ; = { ,200},
6088 <ptm>      ; = { ,500},
6089 <ptm>      ! = { ,100},
6090 <bch>      ? = { ,200},
6091 <ptm>      ? = { ,100},
6092 <ppl>      ? = { ,300},
6093 <pmn>      " = {400,200},
6094 <m-t|ebg|pmn|ppl|ptm> & = {50,50},
6095 <bch>      & = { ,80},

```

```

6096 <cmr>      & = {130,30},
6097 <ugm>      & = {50,100},
6098 <m-t|ebg|pmn> \% = {100, },
6099 <cmr>      \% = {180,50},
6100 <bch>      \% = {50,50},
6101 <ppl|ptm>   \% = {100,100},
6102 <ugm>      \% = {100,50},
6103 <m-t|pmn|ppl> * = {200,200},
6104 <bch>      * = {300,200},
6105 <cmr>      * = {380,20},
6106 <ebg>      * = {500,100},
6107 <ptm|ugm>  * = {400,200},
6108 <m-t|pmn|ppl> + = {150,200},
6109 <cmr>      + = {180,200},
6110 <bch|ugm>  + = {250,250},
6111 <ebg|ptm>  + = {250,200},
6112 <m-t|ebg|pmn|ppl> @ = {50,50},
6113 <bch>      @ = {80,50},
6114 <cmr>      @ = {180,10},
6115 <ptm>      @ = {150,150},
6116 <m-t|bch|ugm> ~ = {150,150},
6117 <cmr|ebg|pmn|ppl|ptm> ~ = {200,150},
6118 <ugm>      {=} = {200,200},
6119 <m-t|bch|ebg|pmn|ppl|ptm|ugm> ( = {200, }, ) = { ,200},
6120 <cmr>      ( = {300, }, ) = { ,70},
6121 <m-t|ebg|ppl|ptm|ugm> / = {100,200},
6122 <cmr>      / = {100,100},
6123 <bch>      / = { ,150},
6124 <pmn>      / = {100,150},
6125 <m-t>      - = {300,300},
6126 <bch|ebg>  - = {300,400},
6127 <pmn>      - = {200,300},
6128 <cmr>      - = {500,300},
6129 <ppl>      - = {300,500},
6130 <ptm>      - = {500,500},
6131 <ugm>      - = {400,700},
6132 <blg>      - = {0,300},
6133 <m-t|pmn>  \textendash = {200,200}, \textendash = {150,150},
6134 <bch>      \textendash = {200,300}, \textendash = {150,200},
6135 <cmr>      \textendash = {500,300}, \textendash = {400,170},
6136 <ebg|ppl|ptm|ugm> \textendash = {300,300}, \textendash = {200,200},
6137 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
6138 <blg>      \textquoteleft = {400,400}, \textquoteright = {400,400},
6139 <cmr>      \textquoteleft = {800,200}, \textquoteright = {800,-20},
6140 <ebg>      \textquoteleft = {800,200}, \textquoteright = {800,200},
6141 <ppl>      \textquoteleft = {700,400}, \textquoteright = {700,400},
6142 <ptm>      \textquoteleft = {800,500}, \textquoteright = {800,500},
6143 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
6144 <blg>      \textquotedblright = {300,300}
6145 <cmr>      \textquotedblleft = {540,100}, \textquotedblright = {500,100}
6146 <ebg>      \textquotedblleft = {700,200}, \textquotedblright = {700,200}
6147 <ppl>      \textquotedblleft = {500,300}, \textquotedblright = {500,300}
6148 <ptm>      \textquotedblleft = {700,400}, \textquotedblright = {700,400}
6149 <ugm>      \textquotedblleft = {600,200}, \textquotedblright = {600,200}
6150 }
6151
6152 <*cmr|pmn>
6153 \SetProtrusion
6154 <cmr> [ name = cmr-it-OT1,
6155 <pmn> [ name = pmnj-it-OT1,
6156 <cmr>   load = cmr-it ]
6157 <pmn>   load = pmnj-it ]
6158 <cmr> { encoding = {OT1,OT4},
6159 <pmn> { encoding = OT1,
6160 <cmr>   family = cmr,

```

```

6161 <pmn>    family   = pmnj,
6162 <cmr>    shape    = it    }
6163 <pmn>    shape    = {it,sl} }
6164 {
6165 <cmr>    \AE = {100,    },
6166 <pmn>    \AE = {    ,-50},
6167 <cmr>    \OE = {100,    },
6168 <pmn>    \OE = {50,     }
6169 <*cmr>
6170 "00 = {200,150}, % \Gamma
6171 "01 = {150,100}, % \Delta
6172 "02 = {150, 50}, % \Theta
6173 "03 = {150, 50}, % \Lambda
6174 "04 = {100,100}, % \Xi
6175 "05 = {100,100}, % \Pi
6176 "06 = {100, 50}, % \Sigma
6177 "07 = {200,150}, % \Upsilon
6178 "08 = {150, 50}, % \Phi
6179 "09 = {150,100}, % \Psi
6180 "0A = { 50, 50} % \Omega
6181 </cmr>
6182 }
6183
6184 </cmr|pmn>
6185 \SetProtrusion
6186 <m-t> [ name      = Tl-it-default,
6187 <bch> [ name      = bch-it-Tl,
6188 <blg> [ name      = blg-it-Tl,
6189 <cmr> [ name      = cmr-it-Tl,
6190 <ebg> [ name      = EBGaramond-it-Tl,
6191 <pmn> [ name      = pmnj-it-Tl,
6192 <ppl> [ name      = ppl-it-Tl,
6193 <ptm> [ name      = ptm-it-Tl,
6194 <ugm> [ name      = ugm-it-Tl,
6195 <m-t> load      = OTl-it ]
6196 <bch> load      = bch-it ]
6197 <blg> load      = blg-Tl ]
6198 <cmr> load      = cmr-it ]
6199 <pmn> load      = pmnj-it ]
6200 <ebg> load      = EBGaramond-it ]
6201 <ppl> load      = ppl-it ]
6202 <ptm> load      = ptm-it ]
6203 <ugm> load      = ugm-it ]
6204 <m-t|bch|cmr|ebg|pmn|ppl> { encoding = {Tl,Ly1},
6205 <blg|ptm|ugm> { encoding = Tl,
6206 <bch>    family   = bch,
6207 <blg>    family   = blg,
6208 <cmr>    family   = cmr,
6209 <pmn>    family   = pmnj,
6210 <ebg>    family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-Of,EBGaramond-TOf},
6211 <ppl>    family   = {ppl,pplx,pplj},
6212 <ptm>    family   = {ptm,ptmx,ptmj},
6213 <ugm>    family   = ugm,
6214 <m-t|bch|ebg|pmn|ppl|ptm> shape    = {it,sl} }
6215 <blg|cmr|ugm> shape    = it    }
6216 {
6217 <m-t|bch|pmn>    _ = {    ,100},
6218 <blg>    _ = {0,300},
6219 <cmr|ugm>    _ = {100,200},
6220 <ebg|ppl|ptm>    _ = {100,100},
6221 <blg>    . = {400,600},
6222 <blg>    {,}= {300,500},
6223 <cmr>    \AE = {100,    },
6224 <pmn>    \AE = {    ,-50},
6225 <bch|pmn>    \OE = { 50,    },

```



```

6226 <cmr> \OE = {100, },
6227 <pmn> 031 = { , -100}, % ff1
6228 <cmr|ptm> 156 = {100, }, % IJ
6229 <ebg> 156 = {50, }, % IJ
6230 <pmn> 156 = {20, }, % IJ
6231 <pmn> 188 = { , -30}, % ij
6232 <pmn> \v t = { , 100},
6233 <m-t|ebg|ppl|ptm> \textbackslash = {100,200},
6234 <cmr|ugm> \textbackslash = {300,300},
6235 <bch> \textbackslash = {150,150},
6236 <pmn> \textbackslash = {100,150},
6237 <ugm> \textbar = {200,200},
6238 <cmr> \textquotedblleft = {500,300},
6239 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
6240 <blg> \textquotedbl = {300,300}, \textquotedblleft = {300,300},
6241 <blg> \textquotedblright = {300,300}, \quotedblbase = {200,600},
6242 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
6243 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
6244 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
6245 <ebg|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
6246 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
6247 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
6248 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
6249 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
6250 <ebg> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
6251 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
6252 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
6253 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
6254 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
6255 <ebg> \guillemotleft = {300,300}, \guillemotright = {200,400},
6256 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
6257 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
6258 <m-t|ebg|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
6259 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
6260 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
6261 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
6262 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { , 200},
6263 <cmr|ebg|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
6264 <bch|pmn> \textless = {100, }, \textgreater = { , 100},
6265 <cmr|ebg|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
6266 <pmn> \textvisiblespace = {100,100}
6267 }
6268
6269 <*m-t|cmr|pmn>
6270 \SetProtrusion
6271 <m-t> [ name = T2A-it-default,
6272 <cmr> [ name = cmr-it-T2A,
6273 <pmn> [ name = pmnj-it-T2A,
6274 <m-t> load = OT1-it ]
6275 <cmr> load = cmr-it ]
6276 <pmn> load = pmnj-it ]
6277 { encoding = T2A,
6278 <cmr> family = cmr,
6279 <pmn> family = pmnj,
6280 <m-t|pmn> shape = {it,sl} }
6281 <cmr> shape = it }
6282 {
6283 <cmr> \CYRA = {100,50},
6284 <pmn> \CYRA = {50, },
6285 <cmr> \CYRB = {50, },
6286 <cmr> \CYRV = {50, },
6287 <pmn> \CYRV = {20,-50},
6288 <cmr> \CYRG = {100, },
6289 <pmn> \CYRG = {10, },
6290 <cmr> \CYRD = {50, },

```

```

6291 <cmr> \CYRE = {50, },
6292 <pmn> \CYRE = {20,-50},
6293 <cmr> \CYRZH = {50, },
6294 <cmr> \CYRZ = {50, },
6295 <pmn> \CYRZ = {20,-50},
6296 <cmr> \CYRI = {50, },
6297 <pmn> \CYRI = { , -30},
6298 <cmr> \CYRISHRT = {50, },
6299 <cmr> \CYRK = {50, },
6300 <pmn> \CYRK = {20, },
6301 <cmr> \CYRL = {50, },
6302 <cmr> \CYRM = {50, },
6303 <pmn> \CYRM = { , -30},
6304 <cmr> \CYRN = {50, },
6305 <cmr> \CYRO = {100, },
6306 <pmn> \CYRO = {50, },
6307 <cmr> \CYRP = {50, },
6308 <cmr> \CYRR = {50, },
6309 <pmn> \CYRR = {20,-50},
6310 <cmr> \CYRS = {100, },
6311 <pmn> \CYRS = {50, },
6312 <cmr> \CYRT = {100, },
6313 <pmn> \CYRT = {70, },
6314 <cmr> \CYRU = {100, },
6315 <pmn> \CYRU = {50, },
6316 <cmr> \CYRF = {100, },
6317 <cmr> \CYRH = {50, },
6318 <cmr> \CYRC = {50, },
6319 <cmr> \CYRCH = {100, },
6320 <cmr> \CYRSH = {50, },
6321 <cmr> \CYRSHCH = {50, },
6322 <cmr> \CYRHRDSN = {100, },
6323 <cmr> \CYRERY = {50, },
6324 <cmr> \CYRSFTSN = {50, },
6325 <cmr> \CYREREV = {50, },
6326 <cmr> \CYRYU = {50, },
6327 <cmr> \CYRYA = {50, },
6328 <pmn> \CYRYA = { , 20},
6329 <pmn> \cyrr = {-50, },
6330 <m-t|pmn> _ = { , 100},
6331 <cmr> _ = {100,200},
6332 <pmn> 031 = { , -100}, % ff1
6333 <pmn> \v t = { , 100},
6334 <m-t> \textbackslash = {100,200}, \quotedblbase = {400,500},
6335 <cmr> \textbackslash = {300,300}, \quotedblbase = {200,600},
6336 <pmn> \textbackslash = {100,150}, \quotedblbase = {150,500},
6337 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
6338 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
6339 <pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
6340 <m-t> \textbraceleft = {200,100}, \textbraceright = {200,200},
6341 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
6342 <pmn> \textbraceleft = {200, }, \textbraceright = { , 200},
6343 <cmr> \textquotedblleft = {500,300},
6344 <cmr> \textless = {300,100}, \textgreater = {200,100}
6345 <pmn> \textless = {100, }, \textgreater = { , 100}
6346 }
6347
6348 </m-t|cmr|pmn>
6349 < *m-t|ptm>
6350 \SetProtrusion
6351 <m-t> [ name = QX-it-default,
6352 <ptm> [ name = ptm-it-QX,
6353 <m-t> load = OT1-it ]
6354 <ptm> load = ptm-it ]
6355 { encoding = {QX},

```

```

6356 <ptm>    family    = {ptm,ptmx,ptmj},
6357          shape      = {it,sl} }
6358    {
6359 <ptm>      009 = { , 50}, % fk
6360          {=} = {100,100},
6361 <m-t>      \textunderscore = {100,100},
6362 <ptm>      \textunderscore = {100,150},
6363          \textbackslash = {100,200},
6364          \quotedblbase = {300,400},
6365 <m-t>      \guillemotleft = {300,300}, \guillemotright = {300,300},
6366 <ptm>      \guillemotleft = {200,400}, \guillemotright = {200,400},
6367          \textexclamdown = {200, }, \textquestiondown = {200, },
6368          \textbraceleft = {200,100}, \textbraceright = {200,200},
6369          \textless = {100,100}, \textgreater = {100,100},
6370          \textminus = {200,200}, \textdegree = {300,150},
6371 <m-t>      \copyright = {100,100}, \textregistered = {100,100}
6372 <ptm>      \textregistered = {100,150}, \copyright = {100,150},
6373 <ptm>      \textDelta = { 70, }, \textdelta = { , 50},
6374 <ptm>      \textpi = { 50, 80}, \textmu = { , 80},
6375 <ptm>      \texteuro = {200, }, \textellipsis = {100,200},
6376 <ptm>      \textquoteleft = {500,400}, \textquoteright = {500,400},
6377 <ptm>      \textquotedblleft = {500,300}, \textquotedblright = {400,400},
6378 <ptm>      \textapprox = { 50, 50}, \textinfty = {100,100},
6379 <ptm>      \textdagger = {150,150}, \textdaggerdbl = {100,100},
6380 <ptm>      \textdiv = {150,150}, \textasciitilde = { 80, 80},
6381 <ptm>      \texttimes = {100,150}, \textpm = { 50, 80},
6382 <ptm>      \textbullet = {300,100}, \textperiodcentered = {300,300},
6383 <ptm>      \textquotesingle = {500,500}, \textquotedbl = {300,300},
6384 <ptm>      \textperthousand = { ,50}
6385    }
6386
6387 </m-t|ptm>
6388 <*cmr|bch>
6389 \SetProtrusion
6390 <cmr> [ name = cmr-it-T5,
6391 <cmr>   load = cmr-it ]
6392 <bch> [ name = bch-it-T5,
6393 <bch>   load = bch-it ]
6394 { encoding = T5,
6395 <bch>   family = bch,
6396 <cmr>   family = cmr,
6397   shape = it }
6398 {
6399 <bch>   _ = { ,100},
6400 <cmr>   _ = {100,200},
6401 <bch>   \textbackslash = {150,150},
6402 <cmr>   \textbackslash = {300,300},
6403 <bch>   \quotesinglbase = {200,500}, \quotedblbase = {150,500},
6404 <cmr>   \quotesinglbase = {300,700}, \quotedblbase = {200,600},
6405 <bch>   \guilsinglleft = {300,400}, \guilsinglright = {200,500},
6406 <cmr>   \guilsinglleft = {500,300}, \guilsinglright = {400,400},
6407 <bch>   \guillemotleft = {200,300}, \guillemotright = {150,400},
6408 <cmr>   \guillemotleft = {400,100}, \guillemotright = {200,300},
6409 <bch>   \textbraceleft = {200, }, \textbraceright = { ,200},
6410 <cmr>   \textbraceleft = {400,100}, \textbraceright = {200,200},
6411 <bch>   \textless = {100, }, \textgreater = { ,100},
6412 <cmr>   \textless = {300,100}, \textgreater = {200,100}
6413 }
6414
6415 </cmr|bch>

```

Slanted is very similar to italic.

```

6416 <*cmr>
6417 \SetProtrusion
6418 [ name = cmr-sl,

```

```

6419     load      = cmr-it-OT1 ]
6420     { encoding = {OT1,OT4},
6421       family   = cmr,
6422       shape    = sl  }
6423     {
6424       L = { ,50},
6425       f = { , -50},
6426       - = {300, },
6427       \textendash = {400, }, \textemdash = {300, }
6428     }
6429
6430 \SetProtrusion
6431 [ name      = cmr-sl-T1,
6432   load      = cmr-it-T1 ]
6433 { encoding = {T1,LY1},
6434   family   = cmr,
6435   shape    = sl  }
6436 {
6437   L = { ,50},
6438   f = { , -50},
6439   - = {300, },
6440   \textendash = {400, }, \textemdash = {300, }
6441 }
6442
6443 \SetProtrusion
6444 [ name      = cmr-sl-T2A,
6445   load      = cmr-it-T2A ]
6446 { encoding = T2A,
6447   family   = cmr,
6448   shape    = sl  }
6449 {
6450   L = { ,50},
6451   f = { , -50},
6452   - = {300, },
6453   \textendash = {400, }, \textemdash = {300, }
6454 }
6455
6456 \SetProtrusion
6457 [ name      = cmr-sl-T5,
6458   load      = cmr-it-T5 ]
6459 { encoding = T5,
6460   family   = cmr,
6461   shape    = sl  }
6462 {
6463   L = { ,50},
6464   f = { , -50},
6465   - = {300, },
6466   \textendash = {400, }, \textemdash = {300, }
6467 }
6468
6469 \SetProtrusion
6470 [ name      = lmr-it-T1,
6471   load      = cmr-it-T1 ]
6472 { encoding = {T1,LY1},
6473   family   = lmr,
6474   shape    = {it,sl} }
6475 {
6476   \textquotedblleft = { ,200}, \textquotedblright = { ,200},
6477   \quotesinglbase    = { ,400}, \quotedblbase      = { ,500}
6478 }
6479

```

Oldstyle numerals are slightly different.

```

6480 \SetProtrusion
6481 [ name = cmr(oldstyle)-it,

```

```

6482     load = cmr-it-T1 ]
6483     { encoding = T1,
6484       family   = {hfor,cmor},
6485       shape    = {it,sl} }
6486     {
6487       1 = {250, 50},
6488       2 = {150,-100},
6489       3 = {100,-50},
6490       4 = {150,150},
6491       6 = {200,  },
6492       7 = {200, 50},
6493       8 = {150,-50},
6494       9 = {100, 50}
6495     }
6496
6497 </cmr>
6498 < *pmn>
6499 \SetProtrusion
6500 [ name   = pmnx-it,
6501   load   = pmnj-it ]
6502 { encoding = OT1,
6503   family   = pmnx,
6504   shape    = {it,sl} }
6505 {
6506   1 = {100,150}
6507 }
6508
6509 \SetProtrusion
6510 [ name   = pmnx-it-T1,
6511   load   = pmnj-it-T1 ]
6512 { encoding = {T1,LY1},
6513   family   = pmnx,
6514   shape    = {it,sl} }
6515 {
6516   1 = {100,150}
6517 }
6518
6519 \SetProtrusion
6520 [ name   = pmnx-it-T2A,
6521   load   = pmnj-it-T2A ]
6522 { encoding = {T2A},
6523   family   = pmnx,
6524   shape    = {it,sl} }
6525 {
6526   1 = {100,150}
6527 }
6528
6529 </pmn>
6530 < *ptm>
6531 \SetProtrusion
6532 [ name   = ptm-it-LY1,
6533   load   = ptm-it-T1 ]
6534 { encoding = {LY1},
6535   family   = {ptm,ptmx,ptmj},
6536   shape    = {it,sl} }
6537 {
6538   -                               = {100,100},
6539   \texttrademark                 = {100,100},
6540   \textregistered                 = {100,100},
6541   \textcopyright                 = {100,100},
6542   \textdegree                    = {300,100},
6543   \textminus                     = {200,200},
6544   \textellipsis                  = {100,200},
6545   \%                             = {  ,  }, % ?
6546   \textcent                      = {100,100},

```

```

6547 \textquotesingle = {500, },
6548 \textflorin = {100, 70},
6549 \textdagger = {150,150},
6550 \textdaggerdbl = {100,100},
6551 \textbullet = {150,150},
6552 \textonesuperior = {150,100},
6553 \texttwosuperior = {150, 50},
6554 \textthreesuperior = {150, 50},
6555 \textparagraph = {100, },
6556 \textperiodcentered = {500,300},
6557 \textonequarter = { 50, },
6558 \textonehalf = { 50, },
6559 \textplusminus = {100,100},
6560 \textmultiply = {150,150},
6561 \textdivide = {150,150}
6562 }
6563
6564 </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).

```

6565 <*(big|ugm)>
6566 \SetProtrusion
6567 <m-t> [ name = OT1-sc,
6568 <bch> [ name = bch-sc,
6569 <cmr> [ name = cmr-sc-OT1,
6570 <ebg> [ name = EBGaramond-sc,
6571 <pmn> [ name = pmnj-sc,
6572 <ppl> [ name = ppl-sc,
6573 <ptm> [ name = ptm-sc,
6574 <m-t> load = default ]
6575 <bch> load = bch-default ]
6576 <cmr> load = cmr-OT1 ]
6577 <ebg> load = EBGaramond-default ]
6578 <pmn> load = pmnj-default ]
6579 <ppl> load = ppl-default ]
6580 <ptm> load = ptm-default ]
6581 <m-t|bch|ebg|pmn> { encoding = OT1,
6582 <cmr|ppl|ptm> { encoding = {OT1,OT4},
6583 <bch> family = bch,
6584 <cmr> family = cmr,
6585 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OfF,EBGaramond-TOfF},
6586 <pmn> family = pmnj,
6587 <ppl> family = {ppl,pplx,pplj},
6588 <ptm> family = {ptm,ptmx,ptmj},
6589 shape = sc }
6590 {
6591 a = {50,50},
6592 <cmr|ebg|ppl|ptm> \ae = {50, },
6593 <bch|pmn> c = {50, },
6594 <bch|ebg|pmn> d = { ,50},
6595 <m-t|bch|cmr|ebg|pmn|ptm> f = { ,50},
6596 <bch|ebg|pmn> g = {50, },
6597 <m-t|cmr|ebg|pmn|ppl|ptm> j = {50, },
6598 <bch> j = {100, },
6599 <m-t|bch|cmr|ebg|pmn|ppl> l = { ,50},
6600 <ptm> l = { ,80},
6601 <m-t|bch|cmr|ebg|pmn|ppl> 013 = { ,50}, % fl
6602 <ptm> 013 = { ,80}, % fl
6603 <bch|ebg|pmn> o = {50,50},

```

```

6604 <ebg|pmn> \oe = {50, },
6605 <ppl> p = { 0, 0},
6606 <bch|ebg|pmn> q = {50,70},
6607 <ppl> q = { 0, },
6608 <m-t|cmr|ebg|pmn|ppl|ptm> r = { , 0},
6609 t = {50,50},
6610 <m-t|bch|cmr|ebg|pmn|ppl> y = {50,50}
6611 <ptm> y = {80,80}
6612 }
6613
6614 \SetProtrusion
6615 <m-t> [ name = T1-sc,
6616 <bch> [ name = bch-sc-T1,
6617 <cmr> [ name = cmr-sc-T1,
6618 <ebg> [ name = EBGaramond-sc-T1,
6619 <pmn> [ name = pmnj-sc-T1,
6620 <ppl> [ name = ppl-sc-T1,
6621 <ptm> [ name = ptm-sc-T1,
6622 <m-t> load = T1-default ]
6623 <bch> load = bch-T1 ]
6624 <cmr> load = cmr-T1 ]
6625 <ebg> load = EBGaramond-T1 ]
6626 <pmn> load = pmnj-T1 ]
6627 <ppl> load = ppl-T1 ]
6628 <ptm> load = ptm-T1 ]
6629 { encoding = {T1,LY1},
6630 <bch> family = bch,
6631 <cmr> family = cmr,
6632 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-Of,EBGaramond-TOf},
6633 <pmn> family = pmnj,
6634 <ppl> family = {ppl,pplx,pplj},
6635 <ptm> family = {ptm,ptmx,ptmj},
6636 shape = sc }
6637 {
6638 a = {50,50},
6639 <cmr|ebg|ppl|ptm> \ae = {50, },
6640 <bch|pmn> c = {50, },
6641 <bch|ebg|pmn> d = { ,50},
6642 <m-t|bch|cmr|ebg|pmn|ptm> f = { ,50},
6643 <bch|ebg|pmn> g = {50, },
6644 <m-t|cmr|ebg|pmn|ppl|ptm> j = {50, },
6645 <bch> j = {100, },
6646 <m-t|bch|cmr|ebg|pmn|ppl> l = { ,50},
6647 <ptm> l = { ,80},
6648 <m-t|bch|cmr|ebg|pmn|ppl> 029 = { ,50}, % fl
6649 <ptm> 029 = { ,80}, % fl
6650 <bch|ebg|pmn> o = {50,50},
6651 <bch|ebg|pmn> \oe = {50, },
6652 <ppl> p = { 0, 0},
6653 <bch|ebg|pmn> q = {50,70},
6654 <ppl> q = { 0, },
6655 <m-t|cmr|ebg|pmn|ppl|ptm> r = { , 0},
6656 t = {50,50},
6657 <m-t|bch|cmr|ebg|pmn|ppl> y = {50,50}
6658 <ptm> y = {80,80}
6659 }
6660
6661 </(blg|ugm)>
6662 <*m-t|cmr>
6663 \SetProtrusion
6664 <m-t> [ name = T2A-sc,
6665 <cmr> [ name = cmr-sc-T2A,
6666 <m-t> load = T2A-default ]
6667 <cmr> load = cmr-T2A ]
6668 { encoding = T2A,

```

```

6669 <cmr>    family    = cmr,
6670    shape      = sc }
6671    {
6672    \cyra = {50,50},
6673    \cyrg = { ,50},
6674    \cyrt = {50,50},
6675    \cyyr = { ,50}
6676    }
6677
6678 </m-t|cmr>
6679 <*m-t>
6680 \SetProtrusion
6681 [ name      = QX-sc,
6682   load      = QX-default ]
6683 { encoding = QX,
6684   shape     = sc }
6685 {
6686   a = {50,50},
6687   f = { ,50},
6688   j = {50, },
6689   l = { ,50},
6690   013 = { ,50}, % fl
6691   r = { , 0},
6692   t = {50,50},
6693   y = {50,50}
6694 }
6695
6696 </m-t>
6697 <*cmr|bch>
6698 \SetProtrusion
6699 <bch> [ name      = bch-sc-T5,
6700 <bch>   load      = bch-T5 ]
6701 <cmr> [ name      = cmr-sc-T5,
6702 <cmr>   load      = cmr-T5 ]
6703 { encoding = T5,
6704 <bch>   family    = bch,
6705 <cmr>   family    = cmr,
6706   shape     = sc }
6707 {
6708   a = {50,50},
6709 <bch>   c = {50, },
6710 <bch>   d = { ,50},
6711   f = { ,50},
6712 <bch>   g = {50, },
6713 <bch>   j = {100, },
6714 <cmr>   j = {50, },
6715   l = { ,50},
6716 <bch>   o = {50,50},
6717 <bch>   q = { 0, },
6718 <cmr>   r = { , 0},
6719   t = {50,50},
6720   y = {50,50}
6721 }
6722
6723 </cmr|bch>
6724 <*pmn>
6725 \SetProtrusion
6726 [ name      = pmnx-sc,
6727   load      = pmnj-sc ]
6728 { encoding = OT1,
6729   family    = pmnx,
6730   shape     = sc }
6731 {
6732   1 = {230,180}
6733 }

```



```

6734
6735 \SetProtrusion
6736 [ name      = pmnx-sc-T1,
6737   load      = pmnj-sc-T1 ]
6738 { encoding = {T1,LY1},
6739   family   = pmnx,
6740   shape     = sc }
6741 {
6742   1 = {230,180}
6743 }
6744

```

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`.

```

6745 \SetProtrusion
6746 [ name      = pmnj-scit,
6747   load      = pmnj-it   ]
6748 { encoding = OT1,
6749   family   = pmnj,
6750   shape     = {scit,si} }
6751 {
6752   a = {50, },
6753   \ae = { , -50},
6754   b = {20, -50},
6755   c = {50, -50},
6756   d = {20, 0},
6757   e = {20, -50},
6758   f = {10, 0},
6759   012 = {10, -50}, % fi
6760   013 = {10, -50}, % fl
6761   014 = {10, -50}, % ffi
6762   015 = {10, -50}, % ffl
6763   g = {50, -50},
6764   i = {20, -50},
6765   j = {20, 0},
6766   k = {20, },
6767   l = {20, 50},
6768   m = { , -30},
6769   n = { , -30},
6770   o = {50, },
6771   \oe = {50, -50},
6772   p = {20, -50},
6773   q = {50, },
6774   r = {20, 0},
6775   s = {20, -30},
6776   t = {70, },
6777   u = {50, -50},
6778   v = {100, },
6779   w = {100, },
6780   y = {50, },
6781   z = { , -50}
6782 }
6783
6784 \SetProtrusion
6785 [ name      = pmnj-scit-T1,
6786   load      = pmnj-it-T1   ]
6787 { encoding = {T1,LY1},
6788   family   = pmnj,
6789   shape     = {scit,si}    }
6790 {
6791   a = {50, },
6792   \ae = { , -50},

```

```

6793     b = {20,-50},
6794     c = {50,-50},
6795     d = {20, 0},
6796     e = {20,-50},
6797     f = {10, 0},
6798     028 = {10,-50}, % fi
6799     029 = {10,-50}, % fl
6800     030 = {10,-50}, % ffi
6801     031 = {10,-50}, % ffl
6802     g = {50,-50},
6803     i = {20,-50},
6804     188 = {20, 0}, % ij
6805     j = {20, 0},
6806     k = {20, },
6807     l = {20,50},
6808     m = { , -30},
6809     n = { , -30},
6810     o = {50, },
6811     \oe = {50,-50},
6812     p = {20,-50},
6813     q = {50, },
6814     r = {20, 0},
6815     s = {20,-30},
6816     t = {70, },
6817     u = {50,-50},
6818     v = {100, },
6819     w = {100, },
6820     y = {50, },
6821     z = { , -50}
6822 }
6823
6824 \SetProtrusion
6825 [ name    = pmnx-scit,
6826   load    = pmnj-scit ]
6827 { encoding = OT1,
6828   family   = pmnx,
6829   shape    = {scit,si} }
6830 {
6831   1 = {100,150}
6832 }
6833
6834 \SetProtrusion
6835 [ name    = pmnx-scit-T1,
6836   load    = pmnj-scit-T1 ]
6837 { encoding = {T1,LY1},
6838   family   = pmnx,
6839   shape    = {scit,si} }
6840 {
6841   1 = {100,150}
6842 }
6843
6844 </pmn>

```

15.8.5 Text companion

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

```

6845 \SetProtrusion
6846 <m-t> [ name    = textcomp ]
6847 <bch> [ name    = bch-textcomp ]
6848 <blg> [ name    = blg-textcomp ]
6849 <cmr> [ name    = cmr-textcomp ]
6850 <ebg> [ name    = EBGaramond-textcomp ]
6851 <pmn> [ name    = pmn-textcomp ]

```

```

6852 <ppl> [ name = ppl-textcomp ]
6853 <ptm> [ name = ptm-textcomp ]
6854 <ugm> [ name = ugm-textcomp ]
6855 <m-t> { encoding = TS1 }
6856 <!m-t> { encoding = TS1,
6857 <bch> family = bch }
6858 <blg> family = blg }
6859 <cmr> family = cmr }
6860 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OfF,EBGaramond-TOfF} }
6861 <pmn> family = {pmnx,pmnj} }
6862 <ppl> family = {ppl,pplx,pplj} }
6863 <ptm> family = {ptm,ptmx,ptmj} }
6864 <ugm> family = ugm }
6865 {
6866 <blg> \textquotestraightbase = {400,500},
6867 <cmr> \textquotestraightbase = {300,300},
6868 <ebg|pmn> \textquotestraightbase = {400,400},
6869 <blg> \textquotestraightdblbase = {300,400},
6870 <cmr|pmn> \textquotestraightdblbase = {300,300},
6871 <ebg> \textquotestraightdblbase = {400,400},
6872 <bch|cmr|ebg|pmn|ugm> \texttwelvewardash = {200,200},
6873 <bch|cmr|ebg|pmn> \textthreequartersemdash = {150,150},
6874 <ugm> \textthreequartersemdash = {200,200},
6875 <blg> \textquotesingle = {500,600},
6876 <cmr|pmn> \textquotesingle = {300,400},
6877 <ebg> \textquotesingle = {400,500},
6878 <ptm> \textquotesingle = {500,500},
6879 <ugm> \textquotesingle = {300,500},
6880 <bch|cmr|pmn> \textasteriskcentered = {200,300},
6881 <blg> \textasteriskcentered = {150,200},
6882 <ebg> \textasteriskcentered = {300,300},
6883 <ugm> \textasteriskcentered = {100,200},
6884 <pmn> \textfactionsolidus = {-200,-200},
6885 <cmr> \textoneoldstyle = {100,100},
6886 <pmn> \textoneoldstyle = { , 50},
6887 <cmr> \textthreeoldstyle = { , 50},
6888 <ebg|pmn> \textthreeoldstyle = { 50, },
6889 <cmr> \textfouroldstyle = { 50, 50},
6890 <ebg|pmn> \textfouroldstyle = { 50, },
6891 <cmr|ebg|pmn> \textsevenoldstyle = { 50, 80},
6892 <cmr> \textlangle = {400, },
6893 <cmr> \textrangle = { ,400},
6894 <m-t|bch|pmn|ptm> \textminus = {200,200},
6895 <cmr|ebg|ppl> \textminus = {300,300},
6896 <blg|ugm> \textminus = {250,300},
6897 <bch|ebg|pmn> \textlbrackdbl = {100, },
6898 <blg> \textlbrackdbl = {200, },
6899 <bch|ebg|pmn> \textrbrackdbl = { ,100},
6900 <blg> \textrbrackdbl = { ,200},
6901 <pmn> \textasciigrave = {200,500},
6902 <bch|blg|cmr|ebg|pmn> \texttildebelow = {200,250},
6903 <pmn> \textasciibreve = {300,400},
6904 <pmn> \textasciicaron = {300,400},
6905 <pmn> \textacutetdbl = {200,300},
6906 <pmn> \textgravedbl = {150,300},
6907 <bch|pmn|ugm> \textdagger = { 80, 80},
6908 <blg> \textdagger = {200,200},
6909 <cmr|ebg> \textdagger = {100,100},
6910 <ptm> \textdagger = {150,150},
6911 <blg> \textdaggerdbl = {150,150},
6912 <cmr|ebg|pmn> \textdaggerdbl = { 80, 80},
6913 <ptm> \textdaggerdbl = {100,100},
6914 <bch> \textbardbl = {100,100},
6915 <blg|ugm> \textbardbl = {150,150},
6916 <bch> \textbullet = {200,200},

```

```

6917 <blg> \textbullet = {400,500},
6918 <cmr|ebg|pmn> \textbullet = { ,100},
6919 <ptm> \textbullet = {150,150},
6920 <ugm> \textbullet = { 50,100},
6921 <bch|cmr|pmn> \textcelsius = { 50, },
6922 <ebg> \textcelsius = { 80, },
6923 <bch> \textflorin = { 50, 50},
6924 <blg> \textflorin = {100,100},
6925 <ebg|ugm> \textflorin = { ,100},
6926 <pmn> \textflorin = { 50,100},
6927 <ptm> \textflorin = { 50, 70},
6928 <cmr> \textcolonmonetary = { , 50},
6929 <ebg|pmn> \textcolonmonetary = { 50, },
6930 <pmn> \textinterrobang = { ,100},
6931 <pmn> \textinterrobangdown = {100, },
6932 <m-t|ebg|ptm> \texttrademark = {100,100},
6933 <bch> \texttrademark = {150,150},
6934 <blg|cmr|ppl> \texttrademark = {200,200},
6935 <pmn> \texttrademark = { 50, 50},
6936 <ugm> \texttrademark = {100,150},
6937 <bch|ugm> \textcent = { 50, },
6938 <ptm> \textcent = {100,100},
6939 <bch> \textsterling = { 50, },
6940 <ugm> \textsterling = { , 50},
6941 <bch> \textbrokenbar = {200,200},
6942 <blg> \textbrokenbar = {250,250},
6943 <ugm> \textbrokenbar = {200,300},
6944 <pmn> \textasciidieresis = {300,400},
6945 <m-t|bch|cmr|ebg|ptm|ugm> \textcopyright = {100,100},
6946 <pmn> \textcopyright = {100,150},
6947 <ppl> \textcopyright = {200,200},
6948 <bch|cmr|ugm> \textordfeminine = {100,200},
6949 <ebg|pmn> \textordfeminine = {200,200},
6950 <bch|cmr|ebg|pmn|ugm> \textlnot = {200, },
6951 <blg> \textlnot = {200,100},
6952 <m-t|bch|cmr|ebg|ptm|ugm> \textregistered = {100,100},
6953 <pmn> \textregistered = { 50,150},
6954 <ppl> \textregistered = {200,200},
6955 <pmn> \textasciimacron = {150,200},
6956 <m-t|ppl|ptm> \textdegree = {300,300},
6957 <bch> \textdegree = {150,200},
6958 <blg|ugm> \textdegree = {200,200},
6959 <cmr|ebg> \textdegree = {400,400},
6960 <pmn> \textdegree = {150,400},
6961 <bch|cmr|ebg|pmn|ugm> \textpm = {150,200},
6962 <blg> \textpm = {100,100},
6963 <ptm> \textpm = { 50, 80},
6964 <bch|blg|ugm> \texttwosuperior = {100,200},
6965 <cmr> \texttwosuperior = { 50,100},
6966 <ebg|pmn> \texttwosuperior = {200,200},
6967 <ptm> \texttwosuperior = { 50, 50},
6968 <bch|blg|ugm> \textthreesuperior = {100,200},
6969 <cmr> \textthreesuperior = { 50,100},
6970 <ebg|pmn> \textthreesuperior = {200,200},
6971 <ptm> \textthreesuperior = { 50, 50},
6972 <pmn> \textasciiacute = {300,400},
6973 <bch|ugm> \textmu = { ,100},
6974 <bch|ebg|pmn> \textparagraph = { ,100},
6975 <bch|cmr|ebg|pmn> \textperiodcentered = {300,400},
6976 <blg> \textperiodcentered = {400,500},
6977 <ptm> \textperiodcentered = {300,300},
6978 <ugm> \textperiodcentered = {200,500},
6979 <bch|blg|ugm> \textonesuperior = {200,300},
6980 <cmr|ebg|pmn> \textonesuperior = {200,200},
6981 <ptm> \textonesuperior = {100,100},

```

```

6982 <bch|ebg|pmn|ugm> \textordmasculine = {200,200},
6983 <blg|cmr> \textordmasculine = {100,200},
6984 <bch|cmr|pmn> \texteuro = {100, },
6985 <ebg> \texteuro = { 50,100},
6986 <bch> \texttimes = {200,200},
6987 <blg|ptm> \texttimes = {100,100},
6988 <cmr> \texttimes = {150,250},
6989 <ebg> \texttimes = {100,150},
6990 <pmn> \texttimes = { 70,100},
6991 <ugm> \texttimes = {200,300},
6992 <bch|ebg|pmn> \textdiv = {150,200}
6993 <blg> \textdiv = {100,100}
6994 <cmr> \textdiv = {150,250}
6995 <ptm> \textdiv = { 50,100},
6996 <ugm> \textdiv = {200,300},
6997 <ptm> \textperthousand = { ,50}
6998 <ugm> \textsection = { ,100},
6999 <ugm> \textonehalf = { 50,100},
7000 <ugm> \textonequarter = { 50,100},
7001 <ugm> \textthreequarters = { 50,100},
7002 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

7003 }
7004
7005 <*cmr|ebg|pmn|ugm>
7006 \SetProtrusion
7007 <cmr> [ name = cmr-textcomp-it ]
7008 <ebg> [ name = EBGaramond-textcomp-it ]
7009 <pmn> [ name = pmn-textcomp-it ]
7010 <ugm> [ name = ugm-textcomp-it ]
7011 { encoding = TS1,
7012 <cmr> family = cmr,
7013 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OfF,EBGaramond-TOfF},
7014 <pmn> family = {pmnx,pmnj},
7015 <ugm> family = ugm,
7016 <!ugm> shape = {it,sl} }
7017 <ugm> shape = it }
7018 {
7019 <cmr> \textquotestraightbase = {300,600},
7020 <ebg|pmn> \textquotestraightbase = {400,400},
7021 <cmr> \textquotestraightdblbase = {300,600},
7022 <ebg> \textquotestraightdblbase = {300,400},
7023 <pmn> \textquotestraightdblbase = {300,300},
7024 \texttwelvewardash = {200,200},
7025 <cmr|ebg|pmn> \textthreequartersemdash = {150,150},
7026 <ugm> \textthreequartersemdash = {200,200},
7027 <cmr> \textquotesingle = {600,300},
7028 <ebg> \textquotesingle = {800,100},
7029 <pmn> \textquotesingle = {300,200},
7030 <ugm> \textquotesingle = {500,500},
7031 <cmr> \textasteriskcentered = {300,200},
7032 <ebg> \textasteriskcentered = {500,100},
7033 <pmn> \textasteriskcentered = {200,300},
7034 <ugm> \textasteriskcentered = {300,150},
7035 <pmn> \textfractionsolidus = {-200,-200},
7036 <cmr> \textoneoldstyle = {100, 50},
7037 <ebg> \textoneoldstyle = {100, },
7038 <pmn> \textoneoldstyle = { 50, },
7039 <ebg> \texttwooldstyle = { 50, },
7040 <pmn> \texttwooldstyle = {-50, },
7041 <cmr> \textthreeoldstyle = {100, 50},
7042 <pmn> \textthreeoldstyle = {-100, },
7043 <cmr> \textfouroldstyle = { 50, 50},
7044 <ebg> \textfouroldstyle = { 50,100},

```

7045	<i>cmr</i>	<code>\textsevenoldstyle</code>	= { 50, 80},
7046	<i>ebg</i>	<code>\textsevenoldstyle</code>	= { 50, },
7047	<i>pmn</i>	<code>\textsevenoldstyle</code>	= { 20, },
7048	<i>cmr</i>	<code>\textlangle</code>	= {400, },
7049	<i>cmr</i>	<code>\textrangle</code>	= { ,400},
7050	<i>cmr ebg</i>	<code>\textminus</code>	= {300,300},
7051	<i>pmn</i>	<code>\textminus</code>	= {200,200},
7052	<i>ugm</i>	<code>\textminus</code>	= {250,300},
7053	<i>ebg pmn</i>	<code>\textlbrackdbl</code>	= {100, },
7054	<i>ebg pmn</i>	<code>\textrbrackdbl</code>	= { ,100},
7055	<i>pmn</i>	<code>\textasciigrave</code>	= {300,300},
7056	<i>cmr ebg pmn</i>	<code>\texttildelow</code>	= {200,250},
7057	<i>pmn</i>	<code>\textasciibreve</code>	= {300,300},
7058	<i>pmn</i>	<code>\textasciicaron</code>	= {300,300},
7059	<i>pmn</i>	<code>\textacutedbl</code>	= {200,300},
7060	<i>pmn</i>	<code>\textgravedbl</code>	= {150,300},
7061	<i>cmr</i>	<code>\textdagger</code>	= {100,100},
7062	<i>ebg</i>	<code>\textdagger</code>	= {200,100},
7063	<i>pmn</i>	<code>\textdagger</code>	= { 80, 50},
7064	<i>ugm</i>	<code>\textdagger</code>	= { 80, 80},
7065	<i>cmr ebg</i>	<code>\textdaggerdbl</code>	= { 80, 80},
7066	<i>pmn</i>	<code>\textdaggerdbl</code>	= { 80, 50},
7067	<i>ugm</i>	<code>\textbardbl</code>	= {150,150},
7068	<i>cmr</i>	<code>\textbullet</code>	= {200,100},
7069	<i>ebg</i>	<code>\textbullet</code>	= {300, },
7070	<i>pmn</i>	<code>\textbullet</code>	= { 30, 70},
7071	<i>ugm</i>	<code>\textbullet</code>	= { 50,100},
7072	<i>cmr</i>	<code>\textcelsius</code>	= {100, },
7073	<i>ebg</i>	<code>\textcelsius</code>	= {200, },
7074	<i>pmn</i>	<code>\textcelsius</code>	= { 50,-50},
7075	<i>ebg</i>	<code>\textflorin</code>	= {100, },
7076	<i>pmn</i>	<code>\textflorin</code>	= { 50,100},
7077	<i>ugm</i>	<code>\textflorin</code>	= { ,100},
7078	<i>cmr</i>	<code>\textcolonmonetary</code>	= {150, },
7079	<i>ebg</i>	<code>\textcolonmonetary</code>	= {100, },
7080	<i>pmn</i>	<code>\textcolonmonetary</code>	= { 50,-50},
7081	<i>cmr ebg</i>	<code>\texttrademark</code>	= {200, },
7082	<i>pmn</i>	<code>\texttrademark</code>	= { 50,100},
7083	<i>ugm</i>	<code>\texttrademark</code>	= {150, 50},
7084	<i>ugm</i>	<code>\textcent</code>	= { 50, },
7085	<i>ugm</i>	<code>\textsterling</code>	= { , 50},
7086	<i>ugm</i>	<code>\textbrokenbar</code>	= {200,300},
7087	<i>pmn</i>	<code>\textasciidieresis</code>	= {300,200},
7088	<i>cmr</i>	<code>\textcopyright</code>	= {100, },
7089	<i>ebg</i>	<code>\textcopyright</code>	= {200,100},
7090	<i>pmn</i>	<code>\textcopyright</code>	= {100,150},
7091	<i>ugm</i>	<code>\textcopyright</code>	= {300, },
7092	<i>cmr</i>	<code>\textordfeminine</code>	= {100,100},
7093	<i>pmn</i>	<code>\textordfeminine</code>	= {200,200},
7094	<i>ugm</i>	<code>\textordfeminine</code>	= {100,200},
7095	<i>cmr ebg</i>	<code>\textlnot</code>	= {300, },
7096	<i>pmn ugm</i>	<code>\textlnot</code>	= {200, },
7097	<i>cmr</i>	<code>\textregistered</code>	= {100, },
7098	<i>ebg</i>	<code>\textregistered</code>	= {200,100},
7099	<i>pmn</i>	<code>\textregistered</code>	= { 50,150},
7100	<i>ugm</i>	<code>\textregistered</code>	= {300, },
7101	<i>pmn</i>	<code>\textasciimacron</code>	= {150,200},
7102	<i>cmr ebg</i>	<code>\textdegree</code>	= {500,100},
7103	<i>pmn</i>	<code>\textdegree</code>	= {150,150},
7104	<i>ugm</i>	<code>\textdegree</code>	= {300,200},
7105	<i>cmr</i>	<code>\textpm</code>	= {150,100},
7106	<i>ebg</i>	<code>\textpm</code>	= {200,150},
7107	<i>pmn ugm</i>	<code>\textpm</code>	= {150,200},
7108	<i>cmr</i>	<code>\textonesuperior</code>	= {400, },
7109	<i>ebg</i>	<code>\textonesuperior</code>	= {300,100},

```

7110 <pmn> \textonesuperior = {200,100},
7111 <ugm> \textonesuperior = {300,300},
7112 <cmr> \texttwosuperior = {400, },
7113 <ebg> \texttwosuperior = {300, },
7114 <pmn> \texttwosuperior = {200,100},
7115 <ugm> \texttwosuperior = {300,200},
7116 <cmr> \textthreesuperior = {400, },
7117 <ebg> \textthreesuperior = {300, },
7118 <pmn> \textthreesuperior = {200,100},
7119 <ugm> \textthreesuperior = {300,200},
7120 <ugm> \textmu = { ,100},
7121 <pmn> \textasciicute = {300,200},
7122 <cmr> \textparagraph = {200, },
7123 <pmn> \textparagraph = { ,100},
7124 <cmr> \textperiodcentered = {500,500},
7125 <ebg|pmn|ugm> \textperiodcentered = {300,400},
7126 <cmr> \textordmasculine = {100,100},
7127 <pmn> \textordmasculine = {200,200},
7128 <ugm> \textordmasculine = {300,200},
7129 <cmr> \texteuro = {200, },
7130 <ebg> \texteuro = {100, },
7131 <pmn> \texteuro = {100,-50},
7132 <cmr> \texttimes = {200,200},
7133 <ebg> \texttimes = {200,100},
7134 <pmn> \texttimes = { 70,100},
7135 <ugm> \texttimes = {200,300},
7136 <cmr|ebg> \textdiv = {200,200},
7137 <pmn> \textdiv = {150,200},
7138 <ugm> \textdiv = {200,300},
7139 <ugm> \textsection = { ,200},
7140 <ugm> \textonehalf = { 50,100},
7141 <ugm> \textonequarter = { 50,100},
7142 <ugm> \textthreequarters = { 50,100},
7143 <ugm> \textsurd = { ,100}
7144 }
7145
7146 </cmr|ebg|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}

```

```

7147 <*cmr>
7148 \SetProtrusion
7149 [ name = cmr-math-letters ]
7150 { encoding = OML,
7151   family = cmm,
7152   series = {m,b},
7153   shape = it }

```

```

7154 {
7155     A = {100, 50}, % \mathnormal
7156     B = { 50,   },
7157     C = { 50,   },
7158     D = { 50, 50},
7159     E = { 50,   },
7160     F = {100, 50},
7161     G = { 50, 50},
7162     H = { 50, 50},
7163     I = { 50, 50},
7164     J = {150, 50},
7165     K = { 50,100},
7166     L = { 50, 50},
7167     M = { 50,   },
7168     N = { 50,   },
7169     O = { 50,   },
7170     P = { 50,   },
7171     Q = { 50, 50},
7172     R = { 50,   },
7173     S = { 50,   },
7174     T = { 50,100},
7175     U = { 50, 50},
7176     V = {100,100},
7177     W = { 50,100},
7178     X = { 50,100},
7179     Y = {100,100},
7180     f = {100,100},
7181     h = {   ,100},
7182     i = {   , 50},
7183     j = {   , 50},
7184     k = {   , 50},
7185     r = {   , 50},
7186     v = {   , 50},
7187     w = {   , 50},
7188     x = {   , 50},
7189     "0B = { 50,100}, % \alpha
7190     "0C = { 50, 50}, % \beta
7191     "0D = {200,150}, % \gamma
7192     "0E = { 50, 50}, % \delta
7193     "0F = { 50, 50}, % \epsilon
7194     "10 = { 50,150}, % \zeta
7195     "12 = { 50,   }, % \theta
7196     "13 = {   ,100}, % \iota
7197     "14 = {   ,100}, % \kappa
7198     "15 = {100, 50}, % \lambda
7199     "16 = {   , 50}, % \mu
7200     "17 = {   , 50}, % \nu
7201     "18 = {   , 50}, % \xi
7202     "19 = { 50,100}, % \pi
7203     "1A = { 50, 50}, % \rho
7204     "1B = {   ,150}, % \sigma
7205     "1C = { 50,150}, % \tau
7206     "1D = { 50, 50}, % \upsilon
7207     "1F = { 50,100}, % \chi
7208     "20 = { 50, 50}, % \psi
7209     "21 = {   , 50}, % \omega
7210     "22 = {   , 50}, % \varepsilon
7211     "23 = {   , 50}, % \vartheta
7212     "24 = {   , 50}, % \varpi
7213     "25 = {100,   }, % \varrho
7214     "26 = {100,100}, % \varsigma
7215     "27 = { 50, 50}, % \varphi
7216     "28 = {100,100}, % \leftharpoonup
7217     "29 = {100,100}, % \leftharpoondown
7218     "2A = {100,100}, % \rightharpoonup

```



```

7219 "2B = {100,100}, % \rightharpoondown
7220 "2C = {300,200}, % \lhook
7221 "2D = {200,300}, % \rhook
7222 "2E = { ,100}, % \triangleright
7223 "2F = {100, }, % \triangleleft
7224 "3A = { ,500}, % ., \ldotp
7225 "3B = { ,500}, % ,
7226 "3C = {200,100}, % <
7227 "3D = {300,400}, % /
7228 "3E = {100,200}, % >
7229 "3F = {200,200}, % \star
7230 "5B = { ,100}, % \flat
7231 "5E = {200,200}, % \smile
7232 "5F = {200,200}, % \frown
7233 "7C = {100, }, % \jmath
7234 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

7235 }
7236

```

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}

```

```

7237 \SetProtrusion
7238 [ name = cmr-math-symbols ]
7239 { encoding = OMS,
7240   family = cmsy,
7241   series = {m,b},
7242   shape = n }
7243 {
7244   A = {150, 50}, % \mathcal
7245   C = { ,100},
7246   D = { , 50},
7247   F = { 50,150},
7248   I = { ,100},
7249   J = {100,150},
7250   K = { ,100},
7251   L = {100, },
7252   M = { 50, 50},
7253   N = { 50,100},
7254   P = { , 50},
7255   Q = { 50, },
7256   R = { , 50},
7257   T = { 50,150},
7258   V = { 50, 50},
7259   W = { , 50},
7260   X = {100,100},
7261   Y = {100, },
7262   Z = {100,150},
7263   "00 = {300,300}, % -
7264   "01 = { ,700}, % \cdot, \cdotp
7265   "02 = {150,250}, % \times
7266   "03 = {150,250}, % *, \ast
7267   "04 = {200,300}, % \div
7268   "05 = {150,250}, % \diamond
7269   "06 = {200,200}, % \pm
7270   "07 = {200,200}, % \mp
7271   "08 = {100,100}, % \oplus
7272   "09 = {100,100}, % \ominus
7273   "0A = {100,100}, % \otimes
7274   "0B = {100,100}, % \oslash
7275   "0C = {100,100}, % \odot
7276   "0D = {100,100}, % \bigcirc

```

```

7277 "0E = {100,100}, % \circ
7278 "0F = {100,100}, % \bullet
7279 "10 = {100,100}, % \asymp
7280 "11 = {100,100}, % \equiv
7281 "12 = {200,100}, % \subseteq
7282 "13 = {100,200}, % \supseteq
7283 "14 = {200,100}, % \leq
7284 "15 = {100,200}, % \geq
7285 "16 = {200,100}, % \preceq
7286 "17 = {100,200}, % \succeq
7287 "18 = {200,200}, % \sim
7288 "19 = {150,150}, % \approx
7289 "1A = {200,100}, % \subset
7290 "1B = {100,200}, % \supset
7291 "1C = {200,100}, % \ll
7292 "1D = {100,200}, % \gg
7293 "1E = {300,100}, % \prec
7294 "1F = {100,300}, % \succ
7295 "20 = {100,200}, % \leftarrow
7296 "21 = {200,100}, % \rightarrow
7297 "22 = {100,100}, % \uparrow
7298 "23 = {100,100}, % \downarrow
7299 "24 = {100,100}, % \leftrightarrows
7300 "25 = {100,100}, % \nearrow
7301 "26 = {100,100}, % \searrow
7302 "27 = {100,100}, % \simeq
7303 "28 = {100,100}, % \Leftarrow
7304 "29 = {100,100}, % \Rightarrow
7305 "2A = {100,100}, % \Uparrow
7306 "2B = {100,100}, % \Downarrow
7307 "2C = {100,100}, % \Leftrightarrow
7308 "2D = {100,100}, % \nrightarrow
7309 "2E = {100,100}, % \swarrow
7310 "2F = { ,100}, % \propto
7311 "30 = { ,400}, % \prime
7312 "31 = {100,100}, % \infty
7313 "32 = {150,100}, % \in
7314 "33 = {100,150}, % \ni
7315 "34 = {100,100}, % \triangle, \bigtriangleup
7316 "35 = {100,100}, % \bigtriangledown
7317 "38 = { ,100}, % \forall
7318 "39 = {100, }, % \exists
7319 "3A = {200, }, % \neg
7320 "3E = {200,200}, % \top
7321 "3F = {200,200}, % \bot, \perp
7322 "5E = {100,200}, % \wedge
7323 "5F = {100,200}, % \vee
7324 "60 = { ,300}, % \vdash
7325 "61 = {300, }, % \dashv
7326 "62 = {100,100}, % \lfloor
7327 "63 = {100,100}, % \rfloor
7328 "64 = {100,100}, % \lceil
7329 "65 = {100,100}, % \rceil
7330 "66 = {150, }, % \lbracket
7331 "67 = { ,150}, % \rbracket
7332 "68 = {400, }, % \langle
7333 "69 = { ,400}, % \rangle
7334 "6C = {100,100}, % \updownarrow
7335 "6D = {100,100}, % \Updownarrow
7336 "6E = {100,300}, % \, \backslash, \setminus
7337 "72 = {100,100}, % \nabla
7338 "79 = {200,200}, % \dagger
7339 "7A = {100,100}, % \ddagger
7340 "7B = {100, }, % \mathparagraph
7341 "7C = {100,100}, % \clubsuit

```

```

7342 "7D = {100,100}, % \diamondsuit
7343 "7E = {100,100}, % \heartsuit
7344 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

7345 }
7346

```

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}
```

```

7347 </cmr>
7348 </cfg-t>

```

15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
7349 <*cfg-u>
```

Symbol font 'a'.

```

7350 <*msa>
7351 \SetProtrusion
7352 [ name = AMS-a ]
7353 { encoding = U,
7354   family = msa }
7355 {
7356   "05 = {150,250}, % \centerdot
7357   "06 = {100,100}, % \lozenge
7358   "07 = { 50, 50}, % \blacklozenge
7359   "08 = { 50, 50}, % \circlearrowright
7360   "09 = { 50, 50}, % \circlearrowleft
7361   "0A = {100,100}, % \rightleftharpoons
7362   "0B = {100,100}, % \leftrightharpoons
7363   "0D = {-50,200}, % \Vdash
7364   "0E = {-50,200}, % \Vvdash
7365   "0F = {-70,150}, % \vDash
7366   "10 = {100,150}, % \twoheadrightarrow
7367   "11 = {100,150}, % \twoheadleftarrow
7368   "12 = { 50,100}, % \leftleftarrows
7369   "13 = { 50, 80}, % \rightrightarrows
7370   "14 = {120,120}, % \upuparrows
7371   "15 = {120,120}, % \downdownarrows
7372   "16 = {200,200}, % \upharpoonright
7373   "17 = {200,200}, % \downharpoonright
7374   "18 = {200,200}, % \upharpoonleft
7375   "19 = {200,200}, % \downharpoonleft
7376   "1A = { 80,100}, % \rightarrowtail
7377   "1B = { 80,100}, % \leftarrowtail
7378   "1C = { 50, 50}, % \leftrightarrows
7379   "1D = { 50, 50}, % \rightleftarrows
7380   "1E = {250, }, % \Lsh
7381   "1F = { ,250}, % \Rsh
7382   "20 = {100,100}, % \rightsquigarrow
7383   "21 = {100,100}, % \leftrightsquigarrow
7384   "22 = {100, 50}, % \looparrowleft
7385   "23 = { 50,100}, % \looparrowright
7386   "24 = { 50, 80}, % \circeq
7387   "25 = { ,100}, % \succsim
7388   "26 = { ,100}, % \gtrsim
7389   "27 = { ,100}, % \gtrapprox
7390   "28 = {150, 50}, % \multimap
7391   "2B = {100,150}, % \doteqdot

```

```

7392 "2C = {100,150}, % \triangleq
7393 "2D = {100, 50}, % \precsim
7394 "2E = {100, 50}, % \lessim
7395 "2F = { 50, 50}, % \lessapprox
7396 "30 = {100, 50}, % \eqslantless
7397 "31 = { 50, 50}, % \eqslantgtr
7398 "32 = {100, 50}, % \curlyeqprec
7399 "33 = { 50,100}, % \curlyeqsucc
7400 "34 = {100, 50}, % \preccurlyeq
7401 "36 = { 50,  }, % \leqslant
7402 "38 = {  , 50}, % \backprime
7403 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
7404 "3C = { 50,100}, % \succcurlyeq
7405 "3E = {  , 50}, % \geqslant
7406 "40 = {  , 50}, % \sqsubset
7407 "41 = { 50,  }, % \sqsupset
7408 "42 = {  ,150}, % \vartriangleright, \rhd
7409 "43 = {150,  }, % \vartriangleleft, \lhd
7410 "44 = {  ,100}, % \trianglerighteq, \unrhd
7411 "45 = {100,  }, % \trianglelefteq, \unlhd
7412 "46 = {100,100}, % \bigstar
7413 "48 = { 50, 50}, % \blacktriangledown
7414 "49 = {  ,100}, % \blacktriangleright
7415 "4A = {100,  }, % \blacktriangleleft
7416 "4B = {  ,150}, % \dashrightarrow (the arrow)
7417 "4C = {150,  }, % \dashleftarrow
7418 "4D = { 50, 50}, % \vartriangle
7419 "4E = { 50, 50}, % \blacktriangle
7420 "4F = { 50, 50}, % \triangledown
7421 "50 = { 50, 50}, % \eqcirc
7422 "56 = {  ,150}, % \rightarrow
7423 "57 = {150,  }, % \leftarrow
7424 "58 = {100,300}, % \checkmark
7425 "5C = { 50, 50}, % \angle
7426 "5D = { 50, 50}, % \measuredangle
7427 "5E = { 50, 50}, % \sphericalangle
7428 "5F = {  , 50}, % \varpropto
7429 "60 = {100,100}, % \smallsmile
7430 "61 = {100,100}, % \smallfrown
7431 "62 = { 50,  }, % \Subset
7432 "63 = {  , 50}, % \Supset
7433 "66 = {150,150}, % \curlywedge
7434 "67 = {150,150}, % \curlyvee
7435 "68 = { 50,150}, % \leftthreetimes
7436 "69 = {100, 50}, % \rightthreetimes
7437 "6C = { 50, 50}, % \bumpeq
7438 "6D = { 50, 50}, % \Bumpeq
7439 "6E = {100,  }, % \lll
7440 "6F = {  ,100}, % \ggg
7441 "70 = { 50,100}, % \ulcorner
7442 "71 = {100, 50}, % \urcorner
7443 "75 = {150,200}, % \dotplus
7444 "76 = { 50,100}, % \backsim
7445 "78 = { 50,100}, % \llcorner
7446 "79 = {100, 50}, % \lrcorner
7447 "7C = {100,100}, % \intercal
7448 "7D = { 50, 50}, % \circledcirc
7449 "7E = { 50, 50}, % \circledast
7450 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

7451 }
7452
7453 (/msa)

```

Symbol font 'b'.

```

7454 (*msb)
7455 \SetProtrusion
7456 [ name      = AMS-b ]
7457 { encoding = U,
7458   family   = msb }
7459 {
7460     A = { 50, 50}, % \mathbb
7461     C = { 50, 50},
7462     G = {   , 50},
7463     L = {   , 50},
7464     P = {   , 50},
7465     R = {   , 50},
7466     T = {   , 50},
7467     V = { 50, 50},
7468     X = { 50, 50},
7469     Y = { 50, 50},
7470     "00 = { 50, 50}, % \lvertneqq
7471     "01 = { 50, 50}, % \gvertneqq
7472     "02 = { 50, 50}, % \nleq
7473     "03 = { 50, 50}, % \ngeq
7474     "04 = {100, 50}, % \nless
7475     "05 = { 50,150}, % \ngtr
7476     "06 = {100, 50}, % \nprec
7477     "07 = { 50,150}, % \nsucc
7478     "08 = { 50, 50}, % \lneqq
7479     "09 = { 50, 50}, % \gneqq
7480     "0A = {100,100}, % \leqslant
7481     "0B = {100,100}, % \geqslant
7482     "0C = {100, 50}, % \lneq
7483     "0D = { 50,100}, % \gneq
7484     "0E = {100, 50}, % \npreceq
7485     "0F = { 50,100}, % \nsucceq
7486     "10 = { 50,   }, % \precnsim
7487     "11 = { 50, 50}, % \succnsim
7488     "12 = { 50, 50}, % \lnsim
7489     "13 = { 50, 50}, % \gnsim
7490     "14 = { 50, 50}, % \lneqq
7491     "15 = { 50, 50}, % \ngeqq
7492     "16 = { 50, 50}, % \precneqq
7493     "17 = { 50, 50}, % \succneqq
7494     "18 = { 50, 50}, % \precnapprox
7495     "19 = { 50, 50}, % \succnapprox
7496     "1A = { 50, 50}, % \lnapprox
7497     "1B = { 50, 50}, % \gnapprox
7498     "1C = {150,200}, % \nsim
7499     "1D = { 50, 50}, % \ncong
7500     "1E = {100,150}, % \diagup
7501     "1F = {100,150}, % \diagdown
7502     "20 = {100, 50}, % \varsubsetneq
7503     "21 = { 50,100}, % \varsupsetneq
7504     "22 = {100, 50}, % \subsetneqq
7505     "23 = { 50,100}, % \supsetneqq
7506     "24 = {100, 50}, % \subseteqq
7507     "25 = { 50,100}, % \supseteqq
7508     "26 = {100, 50}, % \varsubsetneqq
7509     "27 = { 50,100}, % \varsupsetneqq
7510     "28 = {100, 50}, % \subseteqq
7511     "29 = { 50,100}, % \supseteqq
7512     "2A = {100, 50}, % \subsetneq
7513     "2B = { 50,100}, % \supsetneq
7514     "2C = { 50,100}, % \nparallel
7515     "2D = {100,150}, % \nmid
7516     "2E = {150,150}, % \nshortmid
7517     "2F = {100,100}, % \nshortparallel

```

```

7518 "30 = { ,150}, % \nvDash
7519 "31 = { ,150}, % \nVDash
7520 "32 = { ,100}, % \nvDash
7521 "33 = { ,100}, % \nVDash
7522 "34 = { ,100}, % \ntrianglerighteq
7523 "35 = {100, }, % \ntrianglelefteq
7524 "36 = {100, }, % \ntriangleleft
7525 "37 = { ,100}, % \ntriangleright
7526 "38 = {100,200}, % \leftarrow
7527 "39 = {100,200}, % \rightarrow
7528 "3A = {100,100}, % \Leftarrow
7529 "3B = { 50,100}, % \Rightarrow
7530 "3C = {100,100}, % \Leftrightarrow
7531 "3D = {100,200}, % \leftrightarrows
7532 "3E = { 50, 50}, % \divideontimes
7533 "3F = { 50, 50}, % \varnothing
7534 "60 = {200, }, % \Finv
7535 "61 = { , 50}, % \Game
7536 "68 = {100,100}, % \eqsim
7537 "69 = { 50, }, % \beth
7538 "6A = { 50, }, % \gimel
7539 "6B = {150, }, % \daleth
7540 "6C = {200, }, % \lessdot
7541 "6D = { ,200}, % \gtrdot
7542 "6E = {100,200}, % \ltimes
7543 "6F = {150,100}, % \rtimes
7544 "70 = { 50,100}, % \shortmid
7545 "71 = { 50, 50}, % \shortparallel
7546 "72 = {200,300}, % \smallsetminus
7547 "73 = {100,200}, % \thicksim
7548 "74 = { 50,100}, % \thickapprox
7549 "75 = { 50, 50}, % \approx
7550 "76 = { 50,100}, % \succapprox
7551 "77 = { 50, 50}, % \precapprox
7552 "78 = {100,100}, % \curvearrowleft
7553 "79 = { 50,150}, % \curvearrowright
7554 "7A = { 50,200}, % \digamma
7555 "7B = {100, 50}, % \varkappa
7556 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

7557 }
7558
7559 </msb>

```

15.8.8 Euler

Euler Roman font (package `euler`).

```

7560 <*eur>
7561 \SetProtrusion
7562 [ name = euler ]
7563 { encoding = U,
7564   family = eur }
7565 {
7566   "01 = {100,100},
7567   "03 = {100,150},
7568   "06 = { ,100},
7569   "07 = {100,150},
7570   "08 = {100,100},
7571   "0A = {100,100},
7572   "0B = { , 50},
7573   "0C = { ,100},
7574   "0D = {100,100},
7575   "0E = { ,100},

```

```

7576 "0F = {100,100},
7577 "10 = {100,100},
7578 "13 = { ,100},
7579 "14 = { ,100},
7580 "15 = { , 50},
7581 "16 = { , 50},
7582 "17 = { 50,100},
7583 "18 = { 50,100},
7584 "1A = { , 50},
7585 "1B = { , 50},
7586 "1C = { 50,100},
7587 "1D = { 50,100},
7588 "1E = { 50,100},
7589 "1F = { 50,100},
7590 "20 = { , 50},
7591 "21 = { , 50},
7592 "22 = { 50,100},
7593 "24 = { , 50},
7594 "27 = { 50,100},
7595 1 = {100,100},
7596 7 = { 50,100},
7597 "3A = {300,500},
7598 "3B = {200,400},
7599 "3C = {200,100},
7600 "3D = {200,200},
7601 "3E = {100,200},
7602 A = { ,100},
7603 D = { , 50},
7604 J = { 50, },
7605 K = { , 50},
7606 L = { , 50},
7607 Q = { , 50},
7608 T = { 50, },
7609 X = { 50, 50},
7610 Y = { 50, },
7611 h = { , 50},
7612 k = { , 50}
7613 }
7614

```

Extended by the `eulervm` package.

```

7615 \SetProtrusion
7616 [ name = euler-vm,
7617 load = euler ]
7618 { encoding = U,
7619 family = zeur }
7620 {
7621 "28 = {100,200},
7622 "29 = {100,200},
7623 "2A = {100,150},
7624 "2B = {100,150},
7625 "2C = {200,300},
7626 "2D = {200,300},
7627 "2E = { ,100},
7628 "2F = {100, },
7629 "3F = {150,150},
7630 "5B = { ,100},
7631 "5E = {100,100},
7632 "5F = {100,100},
7633 "80 = { , 50},
7634 "81 = {200,250},
7635 "82 = {100,200}
7636 }
7637
7638 </eur>

```

Euler Script font (euca1).

```

7639 (*eus)
7640 \SetProtrusion
7641 [ name      = euscript ]
7642 { encoding = U,
7643   family   = eus  }
7644 {
7645     A = {100,100},
7646     B = { 50,100},
7647     C = { 50, 50},
7648     D = { 50,100},
7649     E = { 50,100},
7650     F = { 50,  },
7651     G = { 50,  },
7652     H = {  ,100},
7653     K = {  , 50},
7654     L = {  ,150},
7655     M = {  , 50},
7656     N = {  , 50},
7657     O = { 50, 50},
7658     P = { 50, 50},
7659     T = {  ,100},
7660     U = {  , 50},
7661     V = { 50, 50},
7662     W = { 50, 50},
7663     X = { 50, 50},
7664     Y = { 50,  },
7665     Z = { 50,100},
7666     "00 = {250,250},
7667     "18 = {200,200},
7668     "3A = {200,150},
7669     "40 = {  ,100},
7670     "5E = {100,100},
7671     "5F = {100,100},
7672     "66 = { 50,  },
7673     "67 = {  , 50},
7674     "6E = {200,200}
7675 }
7676
7677 \SetProtrusion
7678 [ name      = euscript-vm,
7679   load      = euscript ]
7680 { encoding = U,
7681   family   = zeus  }
7682 {
7683     "01 = {600,600},
7684     "02 = {200,200},
7685     "03 = {200,200},
7686     "04 = {200,200},
7687     "05 = {150,150},
7688     "06 = {200,200},
7689     "07 = {200,200},
7690     "08 = {100,100},
7691     "09 = {100,100},
7692     "0A = {100,100},
7693     "0B = {100,100},
7694     "0C = {100,100},
7695     "0D = {100,100},
7696     "0E = {150,150},
7697     "0F = {100,100},
7698     "10 = {150,150},
7699     "11 = {100,100},
7700     "12 = {150,100},
7701     "13 = {100,150},
7702     "14 = {150,100},

```



```
7703 "15 = {100,150},
7704 "16 = {200,100},
7705 "17 = {100,200},
7706 "19 = {150,150},
7707 "1A = {150,100},
7708 "1B = {100,150},
7709 "1C = {100,100},
7710 "1D = {100,100},
7711 "1E = {250,100},
7712 "1F = {100,250},
7713 "20 = {150,200},
7714 "21 = {150,200},
7715 "22 = {150,150},
7716 "23 = {150,150},
7717 "24 = {100,200},
7718 "25 = {150,150},
7719 "26 = {150,150},
7720 "27 = {100,100},
7721 "28 = {100,100},
7722 "29 = {100,150},
7723 "2A = {100,100},
7724 "2B = {100,100},
7725 "2C = {100,100},
7726 "2D = {150,150},
7727 "2E = {150,150},
7728 "2F = {100,100},
7729 "30 = {100,100},
7730 "31 = {100,100},
7731 "32 = {100,100},
7732 "33 = {100,100},
7733 "34 = {100,100},
7734 "35 = {100,100},
7735 "3E = {150,150},
7736 "3F = {150,150},
7737 "60 = { ,200},
7738 "61 = {200, },
7739 "62 = {100,100},
7740 "63 = {100,100},
7741 "64 = {100,100},
7742 "65 = {100,100},
7743 "68 = {300, },
7744 "69 = { ,300},
7745 "6C = {100,100},
7746 "6D = {100,100},
7747 "6F = {100,100},
7748 "72 = {100,100},
7749 "73 = {200,100},
7750 "76 = { ,100},
7751 "77 = {100, },
7752 "78 = { 50, 50},
7753 "79 = {100,100},
7754 "7A = {100,100},
7755 "7D = {150,150},
7756 "7E = {100,100},
7757 "A8 = {100,100},
7758 "A9 = {100,100},
7759 "AB = {200,200},
7760 "BA = { ,200},
7761 "BB = { ,200},
7762 "BD = {200,200},
7763 "DE = {200,200}
7764 }
7765
7766 (/eus)
```

Euler Fraktur font (eufrak).

```

7767 <*euf>
7768 \SetProtrusion
7769 [ name      = mathfrak ]
7770 { encoding = U,
7771   family   = euf  }
7772 {
7773   A = {   , 50},
7774   B = {   , 50},
7775   C = { 50, 50},
7776   D = {   , 80},
7777   E = { 50,   },
7778   G = {   , 50},
7779   L = {   , 80},
7780   O = {   , 50},
7781   T = {   , 80},
7782   X = { 80, 50},
7783   Z = { 80, 50},
7784   b = {   , 50},
7785   c = {   , 50},
7786   k = {   , 50},
7787   p = {   , 50},
7788   q = { 50,   },
7789   v = {   , 50},
7790   w = {   , 50},
7791   x = {   , 50},
7792   1 = {100,100},
7793   2 = { 80, 80},
7794   3 = { 80, 50},
7795   4 = { 80, 50},
7796   7 = { 50, 50},
7797   "12 = {500,500},
7798   "13 = {500,500},
7799   ! = {   ,200},
7800   ' = {200,300},
7801   ( = {200,   },
7802   ) = {   ,200},
7803   * = {200,200},
7804   + = {200,250},
7805   - = {200,200},
7806   {,} = {300,300},
7807   . = {400,400},
7808   {=} = {200,200},
7809   : = {   ,200},
7810   ; = {   ,200},
7811   ] = {   ,200}
7812 }
7813
7814 </euf>
7815 </cfg-u>

```

15.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroi tc) and marvosym²⁴). The euroi tc settings are hidden in the package itself (14.3.7) for ‘free software’ compliance reasons. (Not quite sure whether this is what Karl really had in mind ...)

```

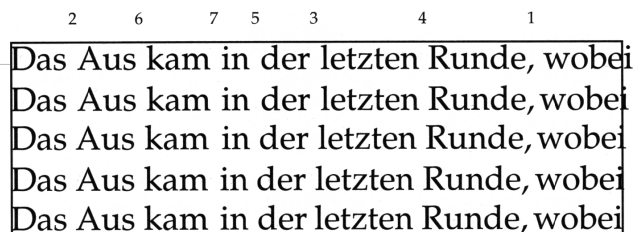
7816 <*cfg-e>
7817 \SetProtrusion
7818 <zpeu> { encoding = U,
7819 <mvs> { encoding = {OT1,U},

```

24 Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

Figure 1:

Example of interword spacing (from: M. Siemoneit, *Typographisches Gestalten*, Frankfurt/M. 1989). The numbers indicate the preference for shrinking the interword space.



```

7820 <zpeu>    family = zpeu }
7821 <mvs>     family = mvs }
7822 {
7823 <zpeu>     E = {50, }
7824 <mvs>     164 = {50,50}, % \EUR
7825 <mvs>     068 = {50,-100} % \EURdig
7826 }
7827
7828 <*zpeu>
7829 \SetProtrusion
7830 { encoding = U,
7831   family = zpeu,
7832   shape = it* }
7833 {
7834   E = {100,-50}
7835 }
7836
7837 \SetProtrusion
7838 { encoding = U,
7839   family = {zpeus,eurosans} }
7840 {
7841   E = {100,50}
7842 }
7843
7844 \SetProtrusion
7845 { encoding = U,
7846   family = {zpeus,eurosans},
7847   shape = it* }
7848 {
7849   E = {200, }
7850 }
7851
7852 </zpeu>
7853 </cfg-e>

```

15.9 Interword spacing

Default unit is space.

```

7854 <*m-t|cmr>
7855 %%% -----
7856 %%% INTERWORD SPACING
7857
7858 </m-t|cmr>
7859 <*m-t>
7860 \SetExtraSpacing
7861 [ name = default ]
7862 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
7863 {

```

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

7864 { , } = { , -500, 500 } ,

- in front of capitals which have optical more room on their left side, e.g., ‘A’, ‘J’, ‘T’, ‘V’, ‘W’, and ‘Y’ [this is not yet possible – RS]
- in front of capitals which have circle/oval shapes on their left side, e.g., ‘C’, ‘G’, ‘O’, and ‘Q’ [ditto – RS]
- after ‘r’ (because of the bigger optical room on the righthand side)

7865 r = { , -300, 300 } ,

- [before or] after lowercase characters with ascenders

7866 b = { , -200, 200 } ,

7867 d = { , -200, 200 } ,

7868 f = { , -200, 200 } ,

7869 h = { , -200, 200 } ,

7870 k = { , -200, 200 } ,

7871 l = { , -200, 200 } ,

7872 t = { , -200, 200 } ,

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

7873 c = { , -100, 100 } ,

7874 p = { , -100, 100 } ,

7875 v = { , -100, 100 } ,

7876 w = { , -100, 100 } ,

7877 z = { , -100, 100 } ,

7878 x = { , -100, 100 } ,

7879 y = { , -100, 100 } ,

- [before or] after lowercase characters with x-height plus descender without additional optical space

7880 i = { , 50, -50 } ,

7881 m = { , 50, -50 } ,

7882 n = { , 50, -50 } ,

7883 u = { , 50, -50 } ,

- after colon and semicolon

7884 : = { , 200, -200 } ,

7885 ; = { , 200, -200 } ,

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

7886 . = { , 250, -250 } ,

7887 ! = { , 250, -250 } ,

7888 ? = { , 250, -250 } ,

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

7889 }

7890

7891 $\langle /m-t \rangle$

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.?

Furthermore, there seems to be a pdfTeX bug with spacing in combination with a non-zero `\spaceskip` (reported by *Axel Berger*):

```
\parfillskip0pt
\rightskip0pt plus 1em
\spaceskip\fontdimen2\font
test test\par
\pdfadjustinterwordglue2
\stbscode\font`t=-50
test test
\bye
```

Some more characters in T2A.²⁵

```
7892 < *cmr >
7893 \SetExtraSpacing
7894   [ name      = T2A,
7895     load      = default ]
7896   { encoding = T2A,
7897     family   = cmr }
7898   {
7899     \cyrg = { , -300, 300 },
7900     \cyrb = { , -200, 200 },
7901     \cyrk = { , -200, 200 },
7902     \cyrs = { , -100, 100 },
7903     \cyrr = { , -100, 100 },
7904     \cyrh = { , -100, 100 },
7905     \cyru = { , -100, 100 },
7906     \cyrt = { , 50, -50 },
7907     \cyrp = { , 50, -50 },
7908     \cyri = { , 50, -50 },
7909     \cyrishrt = { , 50, -50 },
7910   }
7911
```

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 T_EXbook:

‘If the space factor f is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [...] Then the stretch component is multiplied by $f/1000$, while the shrink component is multiplied by $1000/f$.’

The ‘extra space’ (`\fontdimen7`) for Computer Modern Roman is a third of `\fontdimen2`, i.e., 333.

```
7912 \SetExtraSpacing
7913   [ name      = nonfrench-cmr,
7914     load      = default,
7915     context   = nonfrench ]
7916   { encoding = {OT1,T1,LY1,OT4,QX,T5},
```

25 Contributed by *Karl Karlsson*.

```

7917     family    = cmr }
7918   {

```

latex.ltx has:

```

\def\nonfrenchspacing{
  \sfcode`\. 3000
  \sfcode`\? 3000
  \sfcode`\! 3000

```

```

7919   . = {333,2000,-667},
7920   ? = {333,2000,-667},
7921   ! = {333,2000,-667},

```

```

\sfcodes\ : 2000

```

```

7922   : = {333,1000,-500},

```

```

\sfcodes\ ; 1500

```

```

7923   ; = {   , 500,-333},

```

```

\sfcodes\ , 1250

```

```

7924   {,}= {   , 250,-200}

```

```

}

```

```

7925   }

```

```

7926

```

```

7927 </cmr>

```

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.

```

7928 < *m-t >
7929 \SetExtraSpacing
7930 [ name      = nonfrench-default,
7931   load      = default,
7932   context   = nonfrench ]
7933 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
7934 {
7935   . = {240,2000,-667},
7936   ? = {240,2000,-667},
7937   ! = {240,2000,-667},
7938   : = {240,1000,-500},
7939   ; = {   , 500,-333},
7940   {,}= {   , 250,-200}
7941 }
7942

```

Empty settings to prevent spurious warnings.

```

7943 \SetExtraSpacing
7944 [ name = empty ]
7945 { encoding = {TS1} }
7946 { }
7947

```

15.10 Additional kerning

Default unit is 1 em.

```

7948 %%% -----
7949 %%% ADDITIONAL KERNING
7950

```

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

```

7951 \SetExtraKerning
7952   [ name = empty ]
7953   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
7954   { }
7955

```

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 (TeX'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. (Wikipedia²⁶ 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.)

```

7956 \SetExtraKerning
7957   [ name      = french-default,
7958     context   = french,
7959     unit      = space ]
7960   { encoding = {OT1,T1,LY1} }
7961   {
7962     : = {1000,}, % = \fontdimen2
7963     ; = {500, }, % ~ \thinspace
7964     ! = {500, },
7965     ? = {500, }
7966   }
7967

```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTeX.

```

7968 \SetExtraKerning
7969   [ name      = french-guillemets,
7970     context   = french-guillemets,
7971     load      = french-default,
7972     unit      = space ]
7973   { encoding = {T1,LY1} }
7974   {
7975     \guillemotleft = { ,800}, % = 0.8\fontdimen2
7976     \guillemotright = {800, }
7977   }
7978
7979 \SetExtraKerning
7980   [ name      = french-guillemets-OT1,
7981     context   = french-guillemets,
7982     load      = french-default,
7983     unit      = space ]
7984   { encoding = OT1 }
7985   { }
7986

```

15.10.2 Turkish

```

7987 \SetExtraKerning
7988   [ name      = turkish,
7989     context   = turkish ]
7990   { encoding = {OT1,T1,LY1} }

```

26 http://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.

```
7991 {
7992   : = {167, }, % = \thinspace
7993   ! = {167, },
7994   {=} = {167, }
7995 }
7996
7997 </m-t>
7998 </config>
```



[illegible]

[illegible]

```

8167   ж = {ж,ӡ,ӓ},
8168   з = {з,ӓ́},
8169   и = {ӓ́,ӓ́,ӓ́,ӓ́,ӓ́},
8170   к = {к,к,к,к,к,к},
8171   л = {л},
8172   м = {м},
8173   н = {н,н,н,н},
8174   п = {п},
8175   т = {т},
8176   х = {х,х},
8177   ч = {ч,ч,ч,ч},
8178   ш = {ш},
8179   ы = {ӓ́},
8180   э = {ӓ́},
8181   ʼ = {ʼ},
8182   ə = {ə},
8183   ʏ = {ʏ},
8184   Γ = {Γ}, % Greek
8185   Π = {Π}, % Greek
8186   }
8187
8188   % missing: tipa, math, symbols, ...
8189   </CharisSIL>
8190   <Palatino>
8191   \DeclareCharacterInheritance
8192   { encoding = {EU1,EU2,TU},
8193     family = {Palatino} }

```

Unfortunately, I don't have a Palatino variant containing all of the following glyphs. The settings are typeset in T_EX Gyre Pagella; missing glyphs, printed in red, are taken from Charis SIL; glyphs missing even in Charis SIL appear as ‘’. To see the real settings, consult `mt-Palatino.cfg`.

[illegible]

8245 *</Palatino>*8246 **Lato*

8250 { A = {À.Á.Â.Ã.Ä.Å.A}.

8251 $a = \{\grave{a}.\acute{a}.\hat{a}.\tilde{a}.\ddot{a}.\grave{a}.a\}.$

$$C = \{\acute{C}, C\}.$$

8253 $C = \{C.C\}$.

8254 $D = \{D\}$.

8255 $E = \{\dot{E}, \acute{E},$

8256 $e = \{\grave{e}, \acute{e}, \hat{e}, \ddot{e}, \text{e}\},$

8257 $I = \{\hat{I}, \hat{I}, \hat{I}, \hat{I}\},$

8258 $i = \{\grave{i}, \acute{i}, \hat{i}, \ddot{i}, i\}$

8259 $L = \{t\},$

8260 $I = \{t\},$

8261 $N = \{\acute{N}, \acute{N}$

8262 $n = \{\acute{n}, \tilde{n}\},$

8263 $O = \{\emptyset, \grave{O}, \acute{O}, \hat{O}, \tilde{O}, \ddot{O}\},$

8264 $\mathbf{o} = \{\emptyset, \grave{o}, \acute{o}, \hat{o}, \tilde{o}, \ddot{o}\},$

8265 $S = \{\acute{S}, \check{S}\},$

8266 $s = \{\acute{s}, \check{s}\},$

8267 $U = \{\grave{U}, \acute{U}, \hat{U}, \ddot{U}\},$

8268 $u = \{\grave{u}, \acute{u}, \hat{u}, \ddot{u}\},$

8269 $Y = \{\dot{Y}, \ddot{Y}\},$

8270 $y = \{\acute{y}, \ddot{y}\},$

8271 $Z = \{\dot{Z}, \dot{\dot{Z}}, \ddot{\dot{Z}}\},$

8272 $z = \{\acute{z}, \dot{z}, \check{z}\}$

8273 }

8274 */Lato*8275 **FontAwesome*

```
8276 \DeclareCharacterInheritance
```

```
8277 { encoding = {TU,EU1,EU2}.
```

```
8278     family    = {FontAwesome} }
```

8279 { }

```
8280 </FontAwesome>
```

16.2 Character protrusion

```

8281
8282 %%% -----
8283 %%% PROTRUSION
8284
8285 (*LatinModernRoman)
8286 \SetProtrusion
8287 [ name = LMR-default ]
8288 { encoding = {EU1,EU2,TU},
8289   family = Latin Modern Roman }
8290 {
8291   A = {50,50},
8292   Æ = {50, },
8293   F = { ,50},
8294   J = {50, },
8295   K = { ,50},
8296   L = { ,50},
8297   T = {50,50},
8298   V = {50,50},
8299   W = {50,50},
8300   X = {50,50},
8301   Y = {50,50},
8302   k = { ,50},
8303   r = { ,50},
8304   t = { ,70},
8305   v = {50,50},
8306   w = {50,50},
8307   x = {50,50},
8308   y = {50,70},
8309   0 = { ,50},
8310   1 = {100,200},
8311   2 = {50,50},
8312   3 = {50,50},
8313   4 = {70,70},
8314   5 = { ,50},
8315   6 = { ,50},
8316   7 = {50,100},
8317   8 = { ,50},
8318   9 = { ,50},
8319   . = { ,700},
8320   {,}= { ,500},
8321   := { ,500},
8322   ; = { ,500},
8323   ! = { ,100},
8324   ? = { ,200},
8325   @ = {50,50},
8326   ~ = {200,250},
8327   \% = {50,50},
8328   * = {300,300},
8329   + = {250,250},
8330   - = {400,500}, % /hyphen
8331   – = {400,300}, % /endash
8332   — = {300,200}, % /emdash
8333   _ = {200,200}, % /underscore
8334   / = {200,300},
8335   /backslash = {200,300},
8336   ' = {300,400}, % /quotesingle
8337   ‘ = {500,700}, ’ = {500,600},
8338   “ = {500,300}, ” = {200,600},
8339   , = {400,400}, „ = {400,400},
8340   ‹ = {400,400}, › = {300,500},
8341   « = {300,200}, » = {100,400},
8342   ¡ = {100, }, ¿ = {100, },
8343   (= {300, }, ) = { ,300},

```

```

8344 < = {200,100}, > = {100,200},
8345 /braceleft = {400,200}, /braceright = {200,400},
8346 /angleleft = {400, }, /angleright = { },400},
8347 † = {100,100},
8348 ‡ = { 80, 80},
8349 • = {200,200},
8350 · = {400,450}, % / periodcentered
8351 °C = { 80, 50},
8352 ¢ = { , 50},
8353 ° = {400,400},
8354 ™ = {100,200},
8355 © = {100,100},
8356 ® = {100,100},
8357 ª = {100,200},
8358 º = {100,200},
8359 ¹ = {200,250},
8360 º = { 50,100},
8361 ³ = { 50,100},
8362 ¬ = {200, },
8363 − = {300,300},
8364 ± = {150,200},
8365 × = {150,250},
8366 ÷ = {150,250},
8367 € = {100, },
8368 /one.oldstyle = {100,100},
8369 /two.oldstyle = { 50, 50},
8370 /three.oldstyle = { 30, 80},
8371 /four.oldstyle = { 50, 50},
8372 /seven.oldstyle = { 50, 80},
8373 Γ = { ,180}, % /Gamma
8374 Δ = {100,100}, % /Delta
8375 Θ = { 50, 50}, % /Theta
8376 Λ = {100,100}, % /Lambda
8377 % Ξ = {,}, % /Xi
8378 % Π = {,}, % /Pi
8379 Σ = { 50, 50}, % /Sigma
8380 Υ = {100,100}, % /Upsilon
8381 Φ = { 50, 50}, % /Phi
8382 Ψ = { 50, 50}, % /Psi
8383 % Ω = {,}, % /Omega
8384 }
8385
8386 \SetProtrusion
8387 [ name = LMR-it ]
8388 { encoding = {EU1,EU2,TU},
8389 family = Latin Modern Roman,
8390 shape = {it,sl} }
8391 {
8392 A = {125,100},
8393 Æ = {125,-55},
8394 B = {90,-40},
8395 C = {145,-75},
8396 D = {75,-28},
8397 E = {80,-55},
8398 F = {85,-80},
8399 G = {153,-15},
8400 H = {73,-60},
8401 I = {140,-120},
8402 IJ = {140,-80},
8403 J = {135,-80},
8404 K = {70,-30},
8405 L = {87, 40},
8406 M = {67,-45},
8407 N = {75,-55},
8408 O = {150,-30},

```

```

8409    CE = {150,-55},
8410    P = {82,-50},
8411    Q = {150,-30},
8412    R = {75, 15},
8413    S = {90,-65},
8414    $ = {100,-20},
8415    T = {220,-85},
8416    U = {230,-55},
8417    V = {260,-60},
8418    W = {185,-55},
8419    X = {70,-30},
8420    Y = {250,-60},
8421    Z = {90,-60},
8422    a = {150,-10},
8423    b = {170, },
8424    c = {173,-10},
8425    d = {150,-55},
8426    e = {180, },
8427    f = { , -250},
8428    g = {150,-10},
8429    h = {100, },
8430    i = {210, },
8431    ij = {210,-40},
8432    j = { , -40},
8433    k = {110,-50},
8434    l = {240,-110},
8435    m = {80, },
8436    n = {115, },
8437    o = {155, },
8438    q = {170,-40},
8439    r = {155,-40},
8440    s = {130, },
8441    t = {230,-10},
8442    u = {120, },
8443    v = {140,-25},
8444    w = {98,-20},
8445    x = {65,-40},
8446    y = {130,-20},
8447    z = {110,-80},
8448    0 = {170,-85},
8449    1 = {230,110},
8450    2 = {130,-70},
8451    3 = {140,-70},
8452    4 = {130,80},
8453    5 = {160, },
8454    6 = {175,-30},
8455    7 = {250,-150},
8456    8 = {130,-40},
8457    9 = {155,-80},
8458    . = { ,500},
8459    {,}= { ,450},
8460    := { ,300},
8461    ; = { ,300},
8462    & = {130,30},
8463    \% = {180,50},
8464    * = {380,20},
8465    + = {180,200},
8466    @ = {180,10},
8467    ~ = {200,150},
8468    ( = {300, }, ) = { ,70},
8469    / = {100,100},
8470    - = {500,300}, % /hyphen
8471    - = {500,300}, % /endash
8472    — = {400,170}, % /emdash
8473    _ = {100,200}, % /underscore

```



```

8474 ' = {300,400}, % /quotesingle
8475 " = {500,300},
8476 ‘ = {800,200}, ’ = {800,-20},
8477 “ = {540,100}, ” = {500,100},
8478 , = {300,700}, „ = {200,600},
8479 ‹ = {500,300}, › = {400,400},
8480 « = {400,100}, » = {200,300},
8481 ¡ = {200, }, ¿ = {200, },
8482 < = {300,100}, > = {200,100},
8483 /backslash = {300,300},
8484 /braceleft = {400,100}, /braceright = {200,200},
8485 † = {200, 80},
8486 ‡ = {120, 80},
8487 • = {220,100},
8488 · = {550,300}, % / periodcentered
8489 °C = {170, },
8490 ¢ = {100, 50},
8491 ¶ = {200, },
8492 ° = {500,300},
8493 ™ = {200, 70},
8494 © = { 50, 70},
8495 ® = { 50, 70},
8496 º = {140,100},
8497 º = {140,100},
8498 ¹ = {400,150},
8499 º = {250, 80},
8500 ³ = {250, 80},
8501 ¬ = {250, 80},
8502 − = {300,200},
8503 ± = {150,170},
8504 × = {200,200},
8505 ÷ = {200,200},
8506 € = {150, },
8507 /one.oldstyle = {100,100},
8508 /two.oldstyle = {100, 80},
8509 /three.oldstyle = { 80, 50},
8510 /four.oldstyle = { 80, 80},
8511 /five.oldstyle = { 50, },
8512 /six.oldstyle = { 50, },
8513 /seven.oldstyle = { 80, 80},
8514 /eight.oldstyle = { 50, },
8515 Γ = {100,120}, % /Gamma
8516 Δ = {120,100}, % /Delta
8517 Θ = {120, 50}, % /Theta
8518 Λ = {130,100}, % /Lambda
8519 Ξ = {100,}, % /Xi
8520 Π = {100,}, % /Pi
8521 Σ = {100, 50}, % /Sigma
8522 Υ = {180,100}, % /Upsilon
8523 Φ = {130, 70}, % /Phi
8524 Ψ = {130, 50}, % /Psi
8525 Ω = { 50,}, % /Omega
8526 }
8527 </LatinModernRoman>
8528 <*CharisSIL>
8529 \SetProtrusion
8530 [ name = Charis-default ]
8531 { encoding = {EU1,EU2,TU},
8532 family = Charis SIL }
8533 {
8534 A = {50,50},
8535 Æ = {50,50},
8536 C = {50, },
8537 D = { ,50},
8538 F = { ,50},

```

```

8539 G = {50, },
8540 J = {100, },
8541 K = { ,50},
8542 L = { ,50},
8543 Ḷ = { ,100},
8544 O = {50,50},
8545 Œ = {50, },
8546 P = { ,50},
8547 Q = {50,70},
8548 R = { ,50},
8549 ß = { ,40}, % capital sharp s
8550 T = {50,50},
8551 V = {50,50},
8552 W = {50,50},
8553 X = {50,50},
8554 Y = {50,50},
8555 k = { ,50},
8556 ḷ = { ,150},
8557 r = { ,50},
8558 t = { ,50},
8559 v = {50,50},
8560 w = {50,50},
8561 x = {50,50},
8562 y = { ,50},
8563 1 = {150,150},
8564 2 = {50,50},
8565 3 = {50, },
8566 4 = {100,50},
8567 6 = {50, },
8568 7 = {50,80},
8569 9 = {50,50},
8570 . = { ,600},
8571 {,} = { ,500},
8572 : = { ,400},
8573 ; = { ,300},
8574 ! = { ,100},
8575 ? = { ,200},
8576 @ = {50,50},
8577 ~ = {200,250},
8578 \% = { ,50},
8579 * = {300,300},
8580 + = {200,250},
8581 / = { ,200},
8582 /backslash = {150,200},
8583 | = {200,200},
8584 - = {400,500}, % hyphen
8585 - = {200,300}, % endash
8586 — = {150,250}, % emdash
8587 — = {200,200}, % Horizontal Bar = \texttwelveudash
8588 - = {150,150}, % Figure Dash = \textthreequartersemdash
8589 _ = {100,100},
8590 {=} = {100,100},
8591 ‘ = {300,400}, ’ = {300,400},
8592 “ = {300,300}, ” = {300,300},
8593 , = {400,400}, „ = {300,300},
8594 < = {400,300}, > = {300,400},
8595 « = {200,200}, » = {150,300},
8596 ¡ = {100, }, ¿ = {100, },
8597 ( = {200, }, ) = { ,200},
8598 < = {200,150}, > = {100,200},
8599 [ = {100, }, ] = { ,100},
8600 /braceleft = {200, }, /braceright = { ,300},
8601 † = { 80, 80},
8602 ‡ = {100,100},
8603 • = {200,200},

```

8604 ° = {150,200},
8605 ™ = {150,150},
8606 ¢ = { 50, },
8607 £ = { 50, },
8608 ¡ = {200,200},
8609 © = {100,100},
8610 ® = {100,100},
8611 º = {100,200},
8612 ¸ = {200,200},
8613 ¬ = {200, 50},
8614 µ = { ,100},
8615 ¶ = { ,100},
8616 · = {300,400},
8617 ¹ = {200,300},
8618 º = {100,200},
8619 ³ = {100,200},
8620 € = {100, },
8621 ± = {150,200},
8622 × = {200,200},
8623 ÷ = {250,250},
8624 ⁄minus = {200,200},
8625 − = {200,200},
8626 % Cyrillic
8627 Б = { ,50},
8628 Г = { ,130},
8629 Ж = {50,50},
8630 З = {30,50},
8631 Л = {50, },
8632 У = {50,50},
8633 Ф = {50,50},
8634 Ч = {100, },
8635 Ъ = { ,50},
8636 Ь = { ,50},
8637 Э = {50,50},
8638 Ю = { ,40},
8639 Я = {50, },
8640 В = {50,50},
8641 Ё = {50, },
8642 Ъ = {50,100},
8643 Ѓ = {50, },
8644 Љ = {50,50},
8645 Њ = { ,50},
8646 Ћ = {50,50},
8647 Ќ = {100,100},
8648 Ў = {50,50},
8649 Ъ = { ,50},
8650 Ь = { ,50},
8651 Љ = {50,80},
8652 Њ = { ,80},
8653 Ћ = {50,50},
8654 Ќ = {50, },
8655 ЈХ = {50,40},
8656 РХ = { ,50},
8657 ЈЕ = {50, },
8658 ЈЂ = { ,50},
8659 ЈЃ = { ,50},
8660 д = { ,100},
8661 ђ = {50,50},
8662 г = { ,70},
8663 к = { ,50},
8664 л = {50, },
8665 т = {50,50},
8666 ф = {50,50},
8667 ч = {50, },
8668 њ = { ,50},

```

8669   Ъ = { ,50},
8670   Э = { ,50},
8671   Я = {50, },
8672   Ь = {50, },
8673   Њ = { ,50},
8674   Ћ = { ,50},
8675   V = {50,50},
8676   Ў = {50, },
8677   Ъ = { ,50},
8678   Y = {50,50},
8679   Ъ = { ,50},
8680   Ы = { ,50},
8681   Ь = { ,100},
8682   Ƶ = {100,100},
8683   ƶ = {50,50},
8684   Ʒ = {50,70},
8685   ƹ = { ,70},
8686   ƺ = {50,30},
8687   Ы = { ,50},
8688   Ь = { ,50},
8689   %   Д П Ц Ш Щ Ъ Ы Э Ә Ц Ы Ы Э Ә
8690   %   в д ж з и м н п ц ш ы ю ъ е ѝ џ э ә ө ц з д г ь л ж р
8691   % Greek
8692   Δ = {50,50},
8693   Ψ = {50,50},
8694   γ = {70,70},
8695   λ = {40,70},
8696   π = {40,50},
8697   ρ = { ,50},
8698   σ = { ,50},
8699   χ = {50,50},
8700 }
8701
8702 \SetProtrusion
8703   [ name      = Charis-it   ]
8704   { encoding = {EU1,EU2,TU},
8705     family   = Charis SIL,
8706     shape     = {it,sl} }
8707   {
8708   C = {50, },
8709   G = {50, },
8710   J = {50, },
8711   L = {50,50},
8712   O = {50, },
8713   Œ = {50, },
8714   Q = {50, },
8715   S = {50, },
8716   $ = {50, },
8717   T = {70, },
8718   o = {50,50},
8719   p = { ,50},
8720   q = {50, },
8721   t = { ,50},
8722   w = { ,50},
8723   y = { ,50},
8724   1 = {150,100},
8725   3 = {50, },
8726   4 = {100, },
8727   6 = {50, },
8728   7 = {100, },
8729   . = { ,700},
8730   {,} = { ,600},
8731   : = { ,400},
8732   ; = { ,400},
8733   ? = { ,150},

```

```

8734 & = { ,80},
8735 \% = {50,50},
8736 * = {300,200},
8737 + = {250,250},
8738 @ = {80,50},
8739 ~ = {150,150},
8740 / = { ,150},
8741 /backslash = {150,150},
8742 - = {300,400}, % hyphen
8743 – = {200,300}, % endash
8744 — = {150,200}, % emdash
8745 _ = { ,100},
8746 {=} = {200,200},
8747 ± = {150,200},
8748 × = {250,250},
8749 ÷ = {250,250},
8750 ° = {150,200},
8751 · = {300,400},
8752 ‘ = {400,200}, ’ = {400,200},
8753 “ = {300,200}, ” = {400,200},
8754 , = {200,500}, „ = {150,500},
8755 ‹ = {300,400}, › = {200,500},
8756 « = {200,300}, » = {150,400},
8757 ( = {200, }, ) = { ,200},
8758 < = {200,200}, > = {200,200},
8759 /braceleft = {300, }, /braceright = { ,200},
8760 % Cyrillic
8761 Ж = {50,30},
8762 Л = {50, },
8763 У = {50,30},
8764 Ф = {50, },
8765 Ч = {100, },
8766 Ъ = { ,50},
8767 Ь = { ,50},
8768 Э = {50,50},
8769 Я = {50, },
8770 В = {50,50},
8771 Љ = {50,50},
8772 Њ = {140,100},
8773 Ѕ = {70,50},
8774 Ћ = {50,80},
8775 Њ = { ,80},
8776 Ћ = {50,50},
8777 Г = {50,50},
8778 Д = {50,30},
8779 М = {50, },
8780 Ф = {50, },
8781 Ч = {50, },
8782 Ъ = { ,50},
8783 Ь = { ,50},
8784 Э = { ,50},
8785 Я = {50, },
8786 Љ = {50,50},
8787 Њ = { ,50},
8788 В = {50,50},
8789 Ъ = { ,50},
8790 Њ = {140,100},
8791 Ѕ = {70,50},
8792 Ћ = {50,70},
8793 Њ = { ,70},
8794 % Greek
8795 Γ = { ,130},
8796 Δ = {50,50},
8797 Ψ = {50,50},
8798 γ = {70,70},

```

```

8799   λ = {40,70},
8800   π = {40,50},
8801   ρ = { ,50},
8802   σ = { ,50},
8803   χ = {50,50},
8804   }

```

The small caps glyph names in Charis SIL have changed with version 5.0 of the font. We try to get the names right both with LuaT_EX (where we can simply query the font version) and with X_YT_EX (where we check for glyph name).

```

8805
8806 % quick and dirty -- maybe we'll promote this to a
8807 % regular key some time
8808 \define@key{MT@pr@c}{command}{\csname #1\endcsname}
8809
8810 % glyph names have changed with version 5.0 of Charis SIL:
8811 % before: /a.SC, /b.SC, ...
8812 % after:  /a.sc, /b.sc, ...
8813 \ifx\MT@lua\undefined
8814   \gdef\MT@get@CHARIS@SC{
8815     % test whether glyph "a.sc" exists
8816     \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
8817       \gdef\MT@CHARIS@SC{sc}%
8818     \else
8819       \gdef\MT@CHARIS@SC{SC}%
8820     \fi
8821   }
8822 \else
8823   \gdef\MT@get@CHARIS@SC{
8824     \gdef\MT@CHARIS@SC{\MT@lua{
8825       % check font version
8826 % -- why doesn't this work?:
8827 %   f = font.getfont(font.current());
8828 %   i = fontloader.info(f.filename);
8829 %   if (tonumber(i.version) < 5) then;
8830 %     if (tonumber(fontloader.info(font.getfont(font.current()).filename).version) < 5) then;
8831 %       tex.print("SC");
8832 %     else;
8833 %       tex.print("sc");
8834 %     end
8835 %   }}
8836   }
8837 \fi
8838
8839 \SetProtrusion
8840 [ name      = Charis-sc,
8841   load      = Charis-default,
8842   command    = {MT@get@CHARIS@SC} ]
8843 { encoding = {EU1,EU2,TU},
8844   family   = Charis SIL,
8845   shape     = {sc} }
8846 {
8847 % A = {100,100}, % etc., doesn't work with \textsc
8848 /a.\MT@CHARIS@SC = {100,100},
8849 /c.\MT@CHARIS@SC = {50, },
8850 /d.\MT@CHARIS@SC = { ,50},
8851 /f.\MT@CHARIS@SC = { ,50},
8852 /g.\MT@CHARIS@SC = {50, },
8853 /j.\MT@CHARIS@SC = {100, },
8854 /k.\MT@CHARIS@SC = { ,50},
8855 /l.\MT@CHARIS@SC = { ,50},
8856 /f l.\MT@CHARIS@SC = { ,50},
8857 /o.\MT@CHARIS@SC = {50,50},
8858 /oe.\MT@CHARIS@SC = {50, },

```

```

8859 /q.\MT@CHARIS@SC = {50,70},
8860 /r.\MT@CHARIS@SC = { ,50},
8861 /t.\MT@CHARIS@SC = {50,100},
8862 /v.\MT@CHARIS@SC = {50,50},
8863 /w.\MT@CHARIS@SC = {50,50},
8864 /x.\MT@CHARIS@SC = {50,50},
8865 /y.\MT@CHARIS@SC = {50,50}
8866 }
8867 (CharisSIL)
8868 (*Palatino)
8869 \SetProtrusion
8870 [ name = palatino-default ]
8871 { encoding = {EU1,EU2,TU},
8872   family = {Palatino} }
8873 {
8874   A = {50,50},
8875   D = { ,50},
8876   J = {50, },
8877   K = { ,50},
8878   L = { ,50},
8879   O = {25, },
8880   T = {50,50},
8881   V = {50,50},
8882   W = {50,50},
8883   X = {50,50},
8884   Y = {50,50},
8885   b = { ,25},
8886   d = {25,30},
8887   f = { ,50},
8888   g = { ,100},
8889   k = { ,50},
8890   p = { ,50},
8891   q = {50, },
8892   r = { ,50},
8893   t = { ,50}, ◆ = { ,50}, ◆ = { ,50},
8894   v = {75,50},
8895   w = {50,50},
8896   x = {50,50},
8897   y = {50,70},
8898   1 = {100,50},
8899   2 = {25,50},
8900   4 = {50, },
8901   6 = {50, },
8902   9 = {25, },
8903   Æ = {100, },
8904   Œ = {25, },
8905   . = { ,700}, .. = { ,350}, ... = { ,150},
8906   {,} = { ,500},
8907   := { ,500},
8908   ; = { ,500},
8909   ! = { ,100}, !! = { ,100},
8910   ? = { ,200}, ? = { ,200},
8911   @ = {50,50},
8912   ~ = {200,250},
8913   & = {50,100},
8914   \% = {100,100},
8915   * = {200,200},
8916   + = {250,250},
8917   ( = {100, }, ) = { ,300},
8918   / = {200,300},
8919   - = {400,500},
8920   \textendash = {300,300}, \textemdash = {200,200},
8921   \textquoteleft = {500,700}, \textquoteright = {500,700},
8922   \textquotedblleft = {300,400}, \textquotedblright = {300,400},
8923   \textbackslash = {200,300},

```

```

8924 \quotesinglbase = {400,400}, \quotedblbase = {400,400},
8925 \guilsinglleft = {400,400}, \guilsinglright = {300,500},
8926 \guillemotleft = {300,300}, \guillemotright = {200,400},
8927 \textexclamdown = {100, }, \textquestiondown = {100, },
8928 \textbraceleft = {400,200}, \textbraceright = {200,400},
8929 \textless = {200,100}, \textgreater = {100,200},
8930 ≤ = {200,100}, ≥ = {100,200},
8931 \textminus = {300,300},
8932 \texttrademark = {200,200},
8933 \textcopyright = {200,200},
8934 \textregistered = {200,200},
8935 \textdegree = {300,300},
8936 ¡ = {450,500}, ¬ = {250,150},
8937 ¯ = {150,250},
8938 · = {850,700},
8939 ¶ = {100,0},
8940 × = {150,300},
8941 ª = {300,300}, º = {300,300},
8942 ° = {200,400},
8943 ¹ = {400,350}, º = {200,300}, º³ = {250,400},
8944 ⁴ = {250,350}, ⁵ = {200,300}, ⁶ = {250,400},
8945 ⁷ = {200,450}, ⁸ = {250,400}, ⁹ = {200,350},
8946 ⁰ = {200,400},
8947 ¹ = {400,250}, ² = {200,300}, ³ = {250,400},
8948 ⁴ = {250,350}, ⁵ = {200,300}, ⁶ = {250,400},
8949 ⁷ = {200,450}, ⁸ = {250,400}, ⁹ = {200,350},
8950 ± = {150,100}, ÷ = {300,300},
8951 þ = { ,25},
8952 ˙ = {300,450}, ˘ = {300,450},
8953 ˙ = {300,450}, ˘ = {300,450},
8954 † = {200,250}, ‡ = {200,250},
8955 π = {50, },
8956 f = { ,50},
8957 № = {100,150},
8958 \textservicemark = {100,200},
8959 - = {400,500}, - = {400,500}, - = {200,300},
8960 - = {205,305}, — = {200,300}, — = {50,150},
8961 • = {125,200},
8962 % /a.sc = {50,50},
8963 }
8964
8965 \SetProtrusion
8966 [ name = palatino-it ]
8967 { encoding = {EU1,EU2,TU},
8968 family = {Palatino},
8969 shape = {it,sl} }
8970 {
8971 A = {50,50},
8972 Æ = {50, },
8973 B = {50, },
8974 C = {50, },
8975 D = {50,50},
8976 E = {50, },
8977 F = {50, },
8978 G = {50, },
8979 H = {50, },
8980 K = {50, },
8981 L = {50, },
8982 O = {50, },
8983 Œ = {50, },
8984 P = {50, },
8985 Q = {50, },
8986 R = {50, },
8987 S = {50, },
8988 $ = {50, },

```



```

8989 T = {100, },
8990 U = {50, },
8991 V = {100,50},
8992 W = {50, },
8993 X = {50, },
8994 Y = {100,50},
8995 b = { ,50},
8996 c = {25, },
8997 g = {75, },
8998 i = {25, },
8999 m = { ,50},
9000 n = { ,50},
9001 p = { ,25},
9002 q = {25, },
9003 x = { ,50},
9004 1 = {100, },
9005 2 = {50, },
9006 4 = {50, },
9007 7 = {50, },
9008 . = { ,500},   .. = { ,350},   ... = { ,200},
9009 {,} = { ,500},
9010 : = { ,300},
9011 ; = { ,300},
9012 ? = { ,300},   ʔ = { ,300},
9013 & = {50,50},
9014 \% = {100,100},
9015 * = {200,200},
9016 + = {150,200},
9017 @ = {50,50},
9018 ~ = {200,150},
9019 ( = {200, },   ) = { ,200},
9020 / = {100,200},
9021 - = {300,500},
9022 \textendash = {300,300}, \textemdash = {200,200},
9023 \textquoteleft = {700,400}, \textquoteright = {700,400},
9024 \textquotedblleft = {500,300}, \textquotedblright = {500,300},
9025 _ = {100,100},
9026 \textbackslash = {100,200},
9027 \quotesinglbase = {500,500}, \quotedblbase = {400,400},
9028 \guilsinglleft = {400,400}, \guilsinglright = {300,500},
9029 \guillemotleft = {300,300}, \guillemotright = {300,300},
9030 \textexclamdown = {100, }, \textquestiondown = {200, },
9031 \textbraceleft = {200,100}, \textbraceright = {200,200},
9032 \textless = {300,100}, \textgreater = {200,100},
9033 ≤ = {200,100}, ≥ = {100,200},
9034 † = {450,500}, ‡ = {250,150},
9035 · = {850,700},
9036 ¶ = {100,0},
9037 × = {150,300},
9038 ° = {300,250}, ° = {300,300}, ° = {300,250},
9039 º = {300,200},
9040 ¹ = {300,150}, º = {350,200}, º = {250,150},
9041 ² = {350,100}, ³ = {300,50}, ⁴ = {400,100},
9042 ⁵ = {400,50}, ⁶ = {250,50}, ⁷ = {300,50},
9043 ⁸ = {300,300},
9044 ⁹ = {300,350}, ¹ = {300,150}, ² = {250,250},
9045 ³ = {400,200}, ⁴ = {300,100}, ⁵ = {450,200},
9046 ⁶ = {450,150}, ⁷ = {400,250}, ⁸ = {400,200},
9047 ± = {150,100}, ÷ = {300,300},
9048 þ = {50, },
9049 † = {250,200}, ‡ = {250,200},
9050 . = {300,450}, . = {300,450},
9051 ¸ = {300,450}, ¸ = {300,450},
9052 - = {300,500}, - = {300,500}, - = {100,300},
9053 - = {125,305}, - = {200,300}, - = {125,150},

```

```
9054   • = {125,200}
9055   }
9056
9057 \SetProtrusion
9058   [ name      = palatino-sc,
9059     load      = palatino-default ]
9060   { encoding = {EU1,EU2,TU},
9061     family   = {Palatino},
9062     shape    = sc }
9063   {
9064     a = {50,50},
9065     æ = {50, },
9066     b = { 0, 0},
9067     d = { 0, 0},
9068     f = { 0, 0},
9069     g = { 0, 0},
9070     j = {50, },
9071     l = { ,50},
9072     o = { 0, 0},
9073     p = { 0, 0},
9074     q = { 0, },
9075     r = { , 0},
9076     t = {50,50},
9077     y = {50,50},
9078     fl = { 0,50},
9079     ffl = { 0,50},
9080     ◈ = { 0,50},
9081     ◈ = { 0,50}
9082   }
9083 </Palatino>
9084 <Lato> %% No settings yet.
9085 <*FontAwesome>
9086 \SetProtrusion
9087   [ name = empty ]
9088   { encoding = {TU,EU1,EU2},
9089     family   = {FontAwesome} }
9090   { }
9091 </FontAwesome>
9092
```

17 Auxiliary file for micro fine tuning

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

```

9093 < *test>
9094 \documentclass{article}
9095
9096 %% Here you can specify the font you want to test, using
9097 %% the commands \fontfamily, \fontseries and \fontshape.
9098 %% Make sure to end all lines with a comment character!
9099 \newcommand*\TestFont{%
9100   \fontfamily{ppl}%
9101   %% \fontseries{b}%
9102   %% \fontshape{it}% sc, sl
9103 }
9104
9105 \usepackage{ifthen}
9106 \usepackage[T1]{fontenc}
9107 \usepackage[latin1]{inputenc}
9108 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
9109
9110 \pagestyle{empty}
9111 \setlength{\parindent}{0pt}
9112 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
9113 \newcommand*\testprotrusion[2][\crulefill]{%
9114   \ifthenelse{\equal{#1}{r}}{\crulefill}{%
9115     lorem ipsum dolor sit amet,
9116     \ifthenelse{\equal{#1}{l}}{\crulefill}{\leftarrowfill} #2
9117     \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
9118     you know the rest%
9119   \ifthenelse{\equal{#1}{l}}{\crulefill}{%
9120     \linebreak
9121     {\fontencoding{\encodingdefault}%
9122      \fontseries{\seriesdefault}%
9123      \fontshape{\shapedefault}%
9124      \selectfont
9125      Here is the beginning of a line, \dotfill and here is its end}\linebreak
9126 }
9127 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
9128 \def\stripprefix#1>{}
9129 \newcount\charcount
9130 \begin{document}
9131
9132 \microtypesetup{expansion=false}
9133
9134 {\centering The font in this document is called by:\\
9135 \texttt{\showTestFont}\par}\bigskip
9136
9137 \TestFont\selectfont
9138 This line intentionally left empty\linebreak
9139 %% A -- Z
9140 \charcount=65
9141 \loop
9142   \testprotrusion{\char\charcount}
9143   \advance\charcount 1
9144   \ifnum\charcount < 91 \repeat
9145 %% a -- z
9146 \charcount=97
9147 \loop
9148   \testprotrusion{\char\charcount}
9149   \advance\charcount 1
9150   \ifnum\charcount < 123 \repeat
9151 %% 0 -- 9
9152 \charcount=48
9153 \loop

```

```

9154 \testprotrusion{\char\charcount}
9155 \advance\charcount 1
9156 \ifnum\charcount < 58 \repeat
9157 %%
9158 \testprotrusion[r]{,}
9159 \testprotrusion[r]{.}
9160 \testprotrusion[r]{;}
9161 \testprotrusion[r]{:}
9162 \testprotrusion[r]{?}
9163 \testprotrusion[r]{!}
9164 \testprotrusion[l]{\textexclamdown}
9165 \testprotrusion[l]{\textquestiondown}
9166 \testprotrusion[r]{\}}
9167 \testprotrusion[l]{\{ }
9168 \testprotrusion{/}
9169 \testprotrusion{\char~\}
9170 \testprotrusion{-}
9171 \testprotrusion{\textendash}
9172 \testprotrusion{\textemdash}
9173 \testprotrusion{\textquoteleft}
9174 \testprotrusion{\textquoteright}
9175 \testprotrusion{\textquotedblleft}
9176 \testprotrusion{\textquotedblright}
9177 \testprotrusion{\quotesinglbase}
9178 \testprotrusion{\quotedblbase}
9179 \testprotrusion{\guilsinglleft}
9180 \testprotrusion{\guilsinglright}
9181 \testprotrusion{\guillemotleft}
9182 \testprotrusion{\guillemotright}
9183
9184 \newpage
9185 The following displays the current font stretched by 5%,
9186 normal, and shrunk by 5%:
9187
9188 \bigskip
9189 \newlength{\MTln}
9190 \newcommand*{\teststring}
9191 {ABCDEFGH IJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
9192 \settowidth{\MTln}{\teststring}
9193 \microtypesetup{expansion=true}
9194
9195 \parbox{1.05\MTln}{\teststring\linebreak\}
9196 \teststring\par\bigskip
9197 \parbox{0.95\MTln}{\teststring}
9198
9199 \end{document}
9200 /test

```

Needless to say that things may always be improved. For suggestions, mail to w.m.l@gmx.net.

A The title logo

This is `microtype-logo.dtx`. You may treat this file in three different ways:

- compile it by itself
- `\input` it in the body of a `dtx` file
- `\input` it in the preamble: it then provides the command `\printlogo`, which will do just that

The first two cases require the style file `microtype-doc.sty`, which can be generated from `microtype.ins` with:

```
\makefile{microtype-doc.sty}{docsty}
```

9201 *([*logo](#))*

Here's how the logo on the title page was created.³⁰ It has nothing to do with `microtype`, actually, but uses `fontinst`. It is based on an experiment I posted to the [de.comp.text.tex](#) newsgroup.³¹ It will show:

- the character
- the \TeX box
- the bounding box
- kerns

A.1 Macros

To run this file, \TeX needs to find the `afm` file (either in the `TEXINPUTS` path, or in the current working directory). First input `fontinst`.

9202 `\input fontinst.sty`

`bbox.sty` is an addition to `fontinst`, which makes dimensions of the bounding boxes available (and was written by Hàn Thế Thành, by the way). These dimensions are specified in the `afm` file, but not used by \TeX , which is why `fontinst` will discard them otherwise.

9203 `\input bbox.sty`

`\tempdim` Allocate some `dimen` registers.

9204 `\newdimen\tempdim`

`\fboxrulei` Frame width of the box as \TeX sees it.

9205 `\newdimen\fboxrulei`

9206 `\fboxrulei=0.1pt`

`\fboxruleii` Frame width of the bounding box.

9207 `\newdimen\fboxruleii`

9208 `\fboxruleii=0.1pt`

`\kernboxheight` Height of the box indicating the kern.

9209 `\newdimen\kernboxheight`

9210 `\kernboxheight=5pt`

`\scaletoem` An auxiliary macro. Return a dimension relative to the `em`-width of the font. Requires `e-TeX`.

9211 `\setcommand\scaletoem#1{\dimexpr #1 sp*\fontdimen6\font/1000\relax}`

`\showlogo` A `fontinst` incantation whose sole purpose is to produce the logo. Its argument is a string (letters only).

9212 `\fontinstcc`

9213 `\def\showlogo#1{%`

Some fonts do not specify the `\fontdimen6` (width of an `em`) in the `afm` file. In this case, use the font size, which is correct in most cases.

9214 `\ifdim\fontdimen6\font = 0pt`

9215 `\typeout{***-Warning:-no-fontdimen-6-specified-***^^J%}`

9216 `***-setting-it-to-\pdffontsize\font \ifnum\pdfTEXversion < 130 pt\fi-***}`

9217 `\fontdimen6\font=\pdffontsize\font \ifnum\pdfTEXversion < 130 pt\fi\relax`

9218 `\fi`

9219 `\installfonts`

9220 `\input_metrics{{\logofont,\metrics\printbbs{#1}}\relax}`

30 Note that the logo module will not be created when installing `microtype`. Instead, the source file `microtype-logo.dtx` is included as an attachment in the PDF file. If your PDF reader supports this, you can [click here](#) to extract it; alternatively, you may use the `pdftk` tool.

31 Message ID: 42aa3687\$0\$24366\$9b4e6d93@newsread2.arcor-online.net

```

9221 \endinstallfonts
9222 }
9223 \normalcc
      Layers.
9224 \makeatletter
9225 \def\mtl@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
9226 \ifx\mt@objects\undefined\let\mt@objects\@empty\fi
9227 \ifx\mt@order\undefined\let\mt@order\@empty\fi
9228 \xdef\mt@order{\mt@order[(Logo)]}
9229 \let\mtl@resources\@empty
9230 \def\mtl@register#1{%
9231   \immediate\pdfobj{<< /Type/OCG /Name{#1} >>}
9232   \expandafter\xdef\csname mtl@#1\endcsname{\the\pdfobj\space 0 R }
9233   \xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
9234   \xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
9235   \xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
9236 \mtl@register{canvas}
9237 \mtl@register{characters}
9238 \mtl@register{bounding-boxes}
9239 \mtl@register{TeX-boxes}
9240 \xdef\mt@order{\mt@order]}
9241 \global\let\mtl@objects\mt@objects
9242 \def\togglelayer#1#2{%
9243   \pdfstartlink width \wd\logobox height \ht\logobox depth \dp\logobox
9244   user{/Subtype/Link
9245     /BS << /Type/Border/W 0 >> /H/0
9246     /A << /S/SetOCGState
9247     /State[/Toggle \csname mtl@#1\endcsname] >>
9248   }#2\pdfendlink
9249 }

```

\printbbs Preparation.

```

9250 \setcommand\printbbs#1{%
9251   \setbox0\hbox{#1}%
9252   \leavevmode
9253   \kern-\fboxrulei

```

The canvas in the natural width of the text minus protrusion, in color bgcolor.

```

9254 \mtl@layer{canvas}{%
9255   \getboundarychars#1\relax
9256   \tempdim=\dimexpr\wd0 - (\scaletoem{\lcode\font\firstchar}+
9257     \scaletoem{\rcode\font\lastchar})\relax
9258   \kern\dimexpr\scaletoem{\lcode\font\firstchar}\relax
9259   \lower\dimexpr\dp0+0.05em \relax \vbox{\color{bgcolor}%
9260     \hrule width \tempdim
9261     height \dimexpr\dp0+\ht0+0.15em\relax}%
9262   \kern-\tempdim

```

The baseline, in color blcolor.

```

9263   \vbox{\color{blcolor}%
9264     \hrule width \tempdim
9265     height \fboxrulei}%
9266   }%
9267   \kern-\dimexpr\wd0 -\scaletoem{\rcode\font\lastchar}\relax

```

The string.

```

9268 \printbbs #1\relax\relax
9269 }

```

\getboundarychars Get first

```

9270 \def\getboundarychars#1#2\relax{%
9271   \def\firstchar{`#1}%
9272   \getlastchar#1#2\relax
9273 }

```

\getlastchar ... and last character.

```

9274 \def\getlastchar#1#2{%

```

```

9275 \ifx\relax#2\relax
9276 \def\lastchar{`#1}%
9277 \else
9278 \expandafter\getlastchar
9279 \fi #2%
9280 }

```

`\printbss` Loop over all characters of the string.

```

9281 \def\printbss#1#2#3\relax{%
9282 \ifx\relax#1\relax
9283 \else
9284 \ifx\relax#2\relax
9285 \printbb{#1}{}%
9286 \else
9287 \printbb{#1}{#2}%
9288 \fi
9289 \expandafter\printbss
9290 \fi #2#3\relax
9291 }

```

`\printbb` Record the kern between the current and the following character, then print the character. `\kerning` is a fontinst command.

```

9292 \setcommand\printbb#1#2{%
9293 \setbox0\hbox{\kerning{#1}{#2}\xdef\thekern{\number\result}}%
9294 \showboxes{#1}%

```

This could be another application.

```

9295 % \quad
9296 % w: \the\scaletom{\width{#1}},
9297 % bb: \the\scaletom{\bbleft{#1}}/%
9298 % \the\scaletom{\bbright{#1}},
9299 % \the\scaletom{\number\numexpr\width{#1}-\bbright{#1}\relax}
9300 % h: \height{#1}/\bbtop{#1}, \bbbottom{#1}/\depth{#1}\par
9301 }

```

`\showboxes` Print the boxes for char `<#1>`. This won't work if `<#1>` isn't also the PostScript name of the glyph (e.g., 'comma' \neq ',').

```

9302 \setcommand\showboxes#1{%
9303 \leavevmode
9304 \color{texcolor}%

```

We have to record the width of the glyph.

```

9305 \setbox0\hbox{\color{textcolor}{#1}}%
9306 \global\tempdim=\wd0\relax
9307 \kern-\fboxrulei

```

1. *The \TeX box*: Print a frame in color `texcolor`. This frame shows the glyph as \TeX sees it.

```

9308 \mtl@layer{TeX-boxes}{%
9309 \hbox{%
9310 \lower\dimexpr \dp0 + \fboxrulei\relax
9311 \hbox{%
9312 \vbox{%
9313 \hrule height\fboxrulei
9314 \hbox{%
9315 \vrule width\fboxrulei height \dimexpr\ht0 + 2\fboxrulei\relax
9316 \phantom{\unhcopy0}%
9317 \vrule width\fboxrulei
9318 }%
9319 \hrule height\fboxrulei}}}
9320 }%

```

2. *The character*: Now we step back and print the actual glyph. We hold it back until now, so that it will be printed on top of its box.

```

9321 \kern-\wd0
9322 \mtl@layer{characters}{\hbox{\box0}}%

```

Step back by the amount that the character's bounding box differs from the \TeX box on the left side.

```

9323 \kern\dimexpr\scaletom{\bbleft{#1}}-\tempdim-\fboxruleii\relax

```

3. *The bounding box*: will be printed in color `bbcolor`.

```

9324 \mtl@layer{bounding-boxes}{%
9325   {\color{bbcolor}%
9326   \hbox{%
9327     \lower\dimexpr-\scaletom{\bbbottom{#1}}+\fboxruleii\relax
9328     \hbox{%
9329       \vbox{%
9330         \hrule height\fboxruleii
9331         \hbox to \dimexpr\scaletom{\numexpr
9332           \bbright{#1}-\bbleft{#1}\relax}+2\fboxruleii\relax{%
9333           \vrule height \dimexpr\scaletom{\numexpr
9334             \bbtop{#1}-\bbbottom{#1}\relax}%
9335             width\fboxruleii
9336             \hfill
9337             \vrule width\fboxruleii}%
9338           \hrule height\fboxruleii}}}%
9339   }%
9340   \kern-\dimexpr\fboxruleii+\fboxrulei\relax
9341 }%
```

4. *The kern*: We also print a small box in color `kerncolor` indicating the kerning between the current and the next character; filled for negative kerns, empty for positive kerns.

```

9342 \kern\scaletom{\numexpr\width{#1}-\bbright{#1}\relax}%
9343 \mtl@layer{TeX-boxes}{%
9344   {\ifnum\thekern<0
9345     \color{kerncolor}%
9346     \kern\scaletom{\thekern}%
9347     \lower\kernboxheight\hbox{\vrule width -\dimexpr\scaletom{\thekern}\relax
9348       height \kernboxheight}%
9349     \kern\scaletom{\thekern}%
9350   }else
9351     \color{texcolor}%
9352     \ifnum\thekern=0 \else
9353       \lower\kernboxheight
9354       \hbox{%
9355         \vbox{%
9356           \hrule height\fboxrulei
9357           \hbox{%
9358             \vrule height \kernboxheight width\fboxrulei
9359             \kern\dimexpr\scaletom{\thekern}-2\fboxrulei\relax
9360             \vrule width\fboxrulei
9361           }%
9362           \hrule height\fboxruleii}}%
9363       \fi
9364     \fi
9365   }%
9366 }%
9367 % \kern-\fboxrulei
9368 }
```

```

9369 \newbox\logobox
9370 \def\printlogo{%
9371   \setbox\logobox=\hbox{\vbox{%
9372     \MakePercentComment
```

This is the Kepler MM font used in the logo.

```

9373 \def\logofont{pkpri9e10}
9374 \transformfont{\logofont}{\reencodefont{8r}{\fromafm{pkpmmri8a10}}}
9375 \font\thellogofont=\logofont\space at 82pt
```

This would load the italic Palatino font instead.

```

9376 %\def\logofont{pplri}
9377 %\transformfont{\logofont8r}{\reencodefont{8r}{\fromafm{\logofont8a}}}
9378 %\edef\logofont{\logofont8r}
9379 %\font\thellogofont=\logofont\space at 78pt
```

Load the font.


```

9380 \thelogo font
      Protrusion values (overdone for didactic reasons).
9381 \lcode\font`M=96
9382 \rcode\font`e=46
      Now we can generate the logo.
9383 \pdfliteral direct{/SXS gs}%
9384 \showlogo{Microtype}%
9385 % \rlap{\normalfont\normalsize\raisebox{55pt}{\footnotemark[1]}}%
9386 % \kern5pt\[\[3\baselineskip]
9387 % \long\def\@makefnmark##1{%
9388 % \leftskip 0pt
9389 % \parindent 0pt
9390 % \everypar{\parindent 0pt}%
9391 % \leavevmode\hbox to 15pt{\@thefnmark\hss}##1}
9392 % \footnotetext[1]{This graphic display on a
9393 % \togglelayer{canvas}{canvas} the \togglelayer{characters}{characters},
9394 % their \togglelayer{bounding-boxes}{bounding boxes}
9395 % and \togglelayer{TeX-boxes}{\TeX\ boxes}.}
9396 %}%
9397 \edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}
9398 \immediate\pdfobjj<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>%
9399 \immediate\pdfxform
9400 attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}
9401 resources {/Properties <<\mtl@resources>>
9402 /ExtGState << /SXS \the\pdflastobj\space 0 R >> }
9403 \logobox
9404 % \vskip-2.5\baselineskip
9405 % \leavevmode
9406 % \togglelayer{characters}{%
9407 % \pdfrefxform\pdflastxform
9408 % }%
9409 \pdfannot\logodimens{%
9410 /Subtype/Widget /FT/Btn /T(Logo)
9411 %/F 4 % why did I say this?
9412 /AP << /N \the\pdflastxform\space 0 R >>
9413 /AA << /E << /S/SetOCGState /State[/Toggle \mtl@characters] >>
9414 /X << /S/SetOCGState /State[/Toggle \mtl@characters] >>
9415 /D << /S/SetOCGState /State[/Toggle \csname mtl@bounding-boxes\endcsname] >>
9416 /U << /S/SetOCGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >>
9417 >> }%
9418 \vspace{3\baselineskip}
9419 }
      Our font.
9420 \pdfmapline{+pkpmmri8r10 Kep1MM-It_385_575_10_ " TeXBase1Encoding ReEncodeFont " <8r.enc <pkpmmri8a10.pfb}
      Define colours (thered and thegreen are copied from microtype.dtx).
9421 \def\mtdefinecolors{
9422 \definecolor{thered}{rgb}{0.65,0.04,0.07}
9423 \definecolor{thegreen}{rgb}{0.06,0.44,0.08}
9424 \colorlet{texcolor}{thegreen!50} % TeX boxes
9425 \colorlet{kerncolor}{texcolor} % negative kerns
9426 \colorlet{bbcolor}{thered!50} % bounding box
9427 \colorlet{bgcolor}{black!8} % canvas
9428 \colorlet{blcolor}{black!50} % baseline
9429 \colorlet{textcolor}{black!40} % text
9430 }
      Use with microtype.dtx
9431 \ifx\documentclass\@twoclasseserror
9432 \usepackage{xcdraw}{xcolor}
9433 \mtdefinecolors
9434 \else

```

A.2 Document

Now we can start the document.

```

9435 \documentclass[10pt,a4paper]{ltxdoc}
9436 \providecommand\MakePercentComment{\relax}
9437 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}

    Re-use the preamble from microtype.dtx.
9438 \usepackage{microtype-doc}
9439 \usepackage{attachfile}
9440 \makeatletter
9441 \pdfcatalog{/OCProperties << /OCGs [\mt@objects] /D << /Order [\mt@order] >> >>}
9442 \makeatother
9443 \begin{document}

    You are currently reading this.
9444 \DocInput{microtype-logo.dtx}
9445 \newpage
9446   And here it is:
9447 \vfill
9448 \begin{center}
9449   \printlogo \null
9450 \end{center}
9451 \vfill
9452 \expandafter\enddocument
9453 \fi

    That's it.
9454 /logo

```

B The letterspacing illustration

This is `microtype-lssample.dtx`. You may treat this file in three different ways:

- compile it by itself
- `\input` it in the body of a dtx file
- `\input` it in the preamble: it then provides the commands
 - `\lssample`: prints the letterspacing illustration
 - `\anchorarrow`: anchors an arrow for layer `<#1>`
 - `\showarrow`: toggles layer `<#1>` or `<#2>`, and prints `<#2>`

The first two cases require the style file `microtype-doc.sty`, which can be generated from `microtype.ins` with:

```
\makefile{microtype-doc.sty}{docsty}
```

```

9455 \ifx\lssample\undefined
9456 *lssample

```

Upon popular request, here's how I've created the letterspacing illustration.³²

B.1 Macros

Rule width and image height and depth.

```

9457 \makeatletter
9458 \newdimen\lsamount
9459 \newdimen\lsrule
9460 \lsrule=0.2pt
9461 \def\lsheight{8pt}
9462 \def\lsdepth{12pt}

```

³² Note that the `lssample` module will not be created when installing `microtype`. Instead, the source file `microtype-lssample.dtx` is included as an attachment in the PDF file. If your PDF reader supports this, you can [click here](#) to extract it; alternatively, you may use the `pdftk` tool.

Our font (Adobe Caslon).

```
9463 \def\lsfont{\fontfamily{paca}\selectfont}
      Loop over all letters in <#2>, letterspacing them by <#1>.
9464 \def\dols#1#2{\lsamount=#1\relax \dolss#2\enddols}
9465 \def\dolss#1#2\enddols{%
9466   \ifx\empty#2\empty\divide\lsamount 2\fi
9467   \ls{#1}%
9468   \ifx\empty#2\empty\else \dolss#2\enddols \fi
9469 }
```

One tikz picture for each letter.

```
9470 \def\ls#1{%
9471   \begin{tikzpicture}[remember picture,line width=\lsrule]
9472     \tikzstyle{every node}=[inner sep=0pt]
```

The bounding box.

```
9473     \mts@layer{stuff}{%
9474       \node[draw=thegrey,
9475         fill=theshade,
9476         outer sep=\lsrule,
9477         anchor=base,
9478         font=\lsfont]{\phantom{#1}};
9479     }
```

The letter.

```
9480     \node[anchor=base,font=\lsfont](#1){#1};
```

Two auxiliary coordinates.

```
9481     \path (#1.south west) ++(+.5\lsrule,-.5\lsrule) coordinate (#1L);
9482     \path (#1.base east) ++(-.5\lsrule,-\lsdepth) coordinate (#1R);
9483     \mts@layer{stuff}{%
```

Now draw the normal character width,

```
9484       \draw[color=thered!75,
9485         fill=thered!30,
9486         outer sep=\lsrule]
9487         (#1L) rectangle (#1R);
9488       \ifdim\lsamount>0pt
9489         \path (#1.base east) ++(+.5\lsamount,-6pt) coordinate (#1_ls);
9490         \path (#1R) ++(\lsamount+\lsrule,\lsdepth) coordinate (#1E);
```

and the letter space.

```
9491       \draw[color=thered,
9492         fill=thered!50,
9493         outer sep=\lsrule]
9494         (#1R) ++(+\lsrule,+0pt) rectangle (#1E);
9495     \fi
9496   }
9497 \end{tikzpicture}%
9498 \ignorespaces
9499 }
```

Draw the interword space.

```
9500 \def\lssp#1#2#3#4{%
9501   \begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
9502     \mts@layer{stuff}{%
9503       \tikzstyle{every draw}=[anchor=bottom]
9504       \coordinate(#1space) at (#2/2,\lsdepth/2);
9505       \coordinate(#1stretch) at (#2+#3/2,+0pt);
9506       \coordinate(#1shrink) at (#2-#4/2,+0pt);
9507       \draw[color=thegreen,fill=thegreen!50,use as bounding box]
9508         (0,0) rectangle ++(#2,\lsdepth);
9509       \draw[color=thegreen,fill=thegreen!30]
9510         (+#2,-\lsrule) rectangle ++(+#3,-4pt+\lsrule);
9511       \draw[color=thegreen,fill=thegreen!50]
9512         (+#2,-\lsrule) rectangle ++(-#4,-4pt+\lsrule);
9513       \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
```

```

9514         (+#2,-2pt-.5\lsrule) -- ++(+#3,+0pt);
9515         \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!30]
9516         (+#2,-2pt-.5\lsrule) -- ++(-#4,+0pt);
9517     }%
9518 \end{tikzpicture}%
9519 \ignorespaces
9520 }

Layers.
9521 \def\mts@layer#1#2{\pdfliteral page{/OC/#1 BDC}#2\pdfliteral page{EMC}}
9522 \def\mts@layer#1#2{\pdfliteral page{/OC/stuff BDC /OC/#1 BDC}#2\pdfliteral page{EMC EMC}}
9523 \ifx\mt@objects\undefined\let\mt@objects\@empty\fi
9524 \ifx\mt@order\undefined\let\mt@order\@empty\fi
9525 \xdef\mt@order{\mt@order[(Sheep)]}
9526 \let\mts@resources\@empty
9527 \def\mts@register#1{%
9528     \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
9529     \expandafter\xdef\csname mts@#1\endcsname{\the\pdflastobj\space 0 R }
9530     \xdef\mt@objects{\mt@objects\csname mts@#1\endcsname}
9531     \xdef\mt@order{\mt@order\csname mts@#1\endcsname}
9532     \xdef\mts@resources{\mts@resources/#1 \csname mts@#1\endcsname}}
9533 \mts@register{stuff}
9534 \mts@register{tracking}
9535 \mts@register{ispace}
9536 \mts@register{ospace}
9537 \mts@register{istretch}
9538 \mts@register{ishrink}
9539 \mts@register{ostretch}
9540 \mts@register{oshrink}
9541 \mts@register{okern}
9542 \mts@register{ligature}
9543 \mts@register{_compatibility}
9544 \xdef\mt@order{\mt@order]}

Anchor point for the arrow in the code.
9545 \newcommand\anchorarrow[1]{%
9546     \tikz[remember picture,overlay]\node(#1_c){};}

Add an arrow from code to image.
9547 \newcommand\add@arrow[5][left]{%
9548     \tikz[remember picture,overlay,bend angle=14,looseness=0.75,>=latex]{%
9549         \mts@layer{#3}{\draw[->,thick,color=the#2](#4) to[bend #1] (#5);}}%
9550 }

Toggle layer.
9551 \def\toggle@layer#1#2#3{%
9552     \pdfstartlink
9553     user{/Subtype/Link
9554         /BS << /Type/Border/W 0 >> /H/0
9555 %         /BS << /Type/Border/W 1 /S/D /D[4 1] >>
9556 %         /C[0.7 0.7 0.7] /H/0
9557         /Contents(Click to Toggle!)
9558         /A << /S/SetOCGState
9559         /State[/Toggle \csname mts@#1\endcsname] >> }%
9560     \rlap{#2}%
9561     {\fboxsep=0pt \fboxrule=0pt
9562     \mts@layer{stuff}{%
9563         \rlap{\fcolorbox{white}{white}{\vphantom{kg}\color{the#3}#2}}}%
9564     \mts@layer{#1}{%
9565         \fcolorbox{white}{the#3!50}{\vphantom{kg}\color{white}#2}}}%
9566     }%
9567     \pdfendlink
9568 }
9569 \newcommand\showarrow[2][]{%
9570     \ifx\relax#1\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi
9571     \toggle@layer{\@tempa}{\itshape #2}}

```

The environment for our illustration.

```

9572 \def\ls@sample#1{%
9573   \parskip 4pt \parindent 0pt
9574   \par
9575   \vskip4pt
9576   {\leftskip 15pt
9577    \mt@pseudo@margin{\color{theblue}Click on the image to show the kerns
9578     and spacings involved. Click on emphasised words in the text below
9579     to reveal the relation of image and code.\strut}
9580    \mt@layer{_compatibility}%
9581     \mt@place{\rlap{\hskip-\marginparwidth \color{white}%
9582      \vrule width\dimexpr\hsize+\marginparwidth\relax height\mt@unvdimen}}
9583     \mt@pseudo@margin{\color{thered}%
9584      If you had a \acronym{PDF} viewer that understands
9585      \acronym{PDF}\,,{\smaller1.5}, you could hide the arrows selectively.}}
9586     \vskip-\mt@unvdimen}%
9587   \vskip-4pt
9588   \setlength\fbboxsep{4pt}%
9589   \leavevmode
9590   \pdfstartlink
9591     user{/Subtype/Link
9592       /BS << /Type/Border/W 0 >> /H/0
9593       /A << /S/SetOCGState
9594         /State[/Toggle \mts@stuff] >> }%
9595     \fcolorbox{theframe}{theshade}%
9596     {\fontsize{34}{38}\selectfont #1}%
9597   \pdfendlink
9598   \par\medskip
9599   }%
9600   \edef\x{\pdfpageresources{/Properties <<\mts@resources>>}}\x
9601 }

```

Now define the illustration to be used in the document.

```

9602 \def\lssample{%
9603   \ls@sample{%
9604     \dols{0pt}{Stop}
9605     \lssp{o}{0.45em}{0.25em}{0.15em}
9606     \dols{0.16em}{\stealing}\hskip-\dimexpr 0.08em+\lssrule\relax
9607     \lssp{i}{13.82pt}{4.65pt}{2.08pt}
9608     \dols{0.16em}{sheep}
9609     \dols{0pt}{!}
9610   }%

```

Don't forget to add the arrows.

```

9611   \vspace{-\baselineskip}
9612   \add@arrow{red}      {tracking}{lsamount_c.east}{a_ls}
9613   \add@arrow{red}      {okern}   {okernend_c.east}{p_ls}
9614   \add@arrow{green}    {ospace}  {ospace_c.east} {ospace}
9615   \add@arrow{green}    {ispace}  {ispace_c.center}{ispace}
9616   \add@arrow{green!75} {istretch}{istretch_c.east}{istretch.north}
9617   \add@arrow{green!75} {ishrink} {ishrink_c.west} {ishrink.north}
9618   \add@arrow{green!75} {ostretch}{ostretch_c.east}{ostretch.north}
9619   \add@arrow{green!75} {oshrink} {oshrink_c.east} {oshrink.north}
9620   \add@arrow[right]{grey}{ligature}{nolig_c.east} {st.center}
9621 }
9622 \fi

```

This is for use with microtype.dtx

```

9623 \ifx\documentclass\@twoclasseserror
9624   \usepackage{tikz}
9625 \else

```

B.2 Document

```

9626 \documentclass[10pt,a4paper]{ltxdoc}
9627 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}

```

Re-use the preamble from microtype.dtx.

```

9628 \usepackage{microtype-doc}
9629 \usepackage{attachfile}
9630 \usepackage{tikz}
9631 \makeatletter
9632 \pdfcatalog{/OCProperties << /OCGs [\mt@objects]
9633                               /D << /Order [\mt@order] /BaseState/OFF >> >> }
9634 \makeatother
9635 \begin{document}
  You are currently reading this.
9636 \DocInput{microtype-lssample.dtx}
  Now show what we are able to do.
9637 \noindent
9638 Since a picture is worth a thousand words, probably even more if, in our
9639 case, it depicts a couple of letterspaced words, let's bring one to sum up
9640 these somewhat confusing options. Suppose you had the following settings
9641 (which I would in no way recommend; they are only for illustrative purposes):
9642 \begin{verbatim}
9643 \SetTracking
9644   [ no ligatures = {"\anchorarrow{nolig}"f},
9645     spacing      = {60"\anchorarrow{ispace}"0*, "%
9646                   "-1"\anchorarrow{istretch}"00*, "\anchorarrow{ishrink}"},
9647     outer spacing = {4"\anchorarrow{ospace}"50, "%
9648                   "2"\anchorarrow{ostretch}"50, 1"\anchorarrow{oshrink}"50},
9649     outer kerning = {"\anchorarrow{okernbegin}"*, "%
9650                   "\anchorarrow{okernend}"*} ]
9651   { encoding = * }
9652   { 1"\anchorarrow{lsamount}"60 }
9653 \end{verbatim}
9654 and then write:
9655 \begin{verbatim}
9656 Stop \textls{stealing sheep}!
9657 \end{verbatim}
9658 this is the (typographically dubious) outcome:
9659
9660 \lssample
9661
9662 \noindent
9663 While the word 'Stop' is not letterspaced, the space between the letters in
9664 the other two words is expanded by the \showarrow[tracking]{tracking-amount}{red}
9665 of 160/1000\,em\,=\allowbreak\,0.16\,em.
9666 The \showarrow[ispace]{inner~space}{green} within the letterspaced text is
9667 increased by 60\%, while its \showarrow[istretch]{stretch}{green} amount is
9668 decreased by 10\% and the \showarrow[ishrink]{shrink}{green} amount is left
9669 untouched.
9670 The \showarrow[ospace]{outer~space}{green} (of 0.45\,em) immediately before the
9671 piece of text may \showarrow[ostretch]{stretch}{green} by 0.25\,em and
9672 \showarrow[oshrink]{shrink}{green} by 0.15\,em.
9673 Note that there is no outer space after the text, since the exclamation mark
9674 immediately follows; instead, the default \showarrow[okern]{outer~kern}{red}
9675 of half the letterspace amount (0.08\,em) is added.
9676 Furthermore, one \showarrow{ligature}{grey} wasn't broken up, because we
9677 neglected to specify the '~|s|' in the |no ligatures| key.
9678
9679 \expandafter\enddocument
9680 \fi
9681 \lssample

```

C Change history

2004/09/11 Version 1.0

General: Initial version 1

2004/09/21 Version 1.1

General: configuration file names in lowercase (suggested by <i>Harald Harders</i>)	87	list	89
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i>)	147	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i>)	45
Protrusion: add factors for some more characters	154	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i>)	94
settings for Adobe Minion (contributed by <i>Harald Harders</i>)	155	<code>\MT@pdftex@no</code> : fix: version check (reported by <i>Harald Harders</i>)	40
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance	120	<code>\MT@permute</code> : don't use sets for empty encoding	121
<code>\MT@declare@sets</code> : remove spaces around set name	106	<code>\MT@setup@expansion</code> : issue an error instead of a warning, when pdfTeX version is too old for <code>autoexpand</code>	137
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded	87	<code>\MT@split@codes</code> : fix: allow zero and negative values	63
<code>\MT@get@basefamily</code> : only remove suffixes 'x' or 'j'	88	<code>\MT@use@set</code> : remove spaces around set name	110
<code>\MT@get@listname@</code> : don't check for empty attributes			

2004/10/03 Version 1.2

Font aliases: declare <code>cmor</code> as an alias of <code>cmr</code>	144	<code>\MT@get@inh@list</code> : fix: set inheritance list \globally to <code>\@empty</code>	91
Font sets: new: <code>allmath</code> and <code>basicmath</code>	143	<code>\MT@get@listname@</code> : alternatively check for alias font name	89
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding	178	<code>\MT@get@size</code> : additional magic to catch some errors	108
add settings for Computer Modern Roman math symbols	183	<code>\MT@get@size@</code> : hijack <code>\set@fontsize</code> instead of <code>\@setfontsize</code>	108
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement	60	<code>\MT@loop</code> : fix: new macro, used instead of <code>\loop</code>	49
<code>\MT@get@basefamily</code> : also remove 'w' (swash capitals)	88	<code>\MT@maybe@do</code> : also check for alias font name	60
<code>\MT@get@highlevel</code> : check whether defaults have changed	106	<code>\MT@permute@@@@</code> : more sanity checks for <code>\SetProtrusion</code> and <code>\SetExpansion</code>	123
		<code>\MT@setupfont</code> : also search for alias font file	57
		fix: call <code>\@enc@update</code> if necessary	57

2004/10/27 Version 1.3

General: fix: specifying <code>load</code> option does no longer require to give a name, too	116	<code>\MT@fix@catcode</code> : check some category codes (compatibility with german)	35
Font aliases: declare <code>aer</code> , <code>zer</code> and <code>hfor</code> as aliases of <code>cmr</code>	144	<code>\MT@load@list</code> : check whether list exists	87

2004/11/12 Version 1.4

General: check for <code>pdfcpot</code>	54	(OT1, T1, lmr)	160
don't use scratch registers in global definitions	91	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options	131
use <code>\pickup@font</code> instead of <code>\define@newfont</code> as the hook for <code>\MT@setupfont</code>	100	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too	113
use one instead of five counters	50		
Protrusion: tweak quote characters for <code>cmr</code> variants			

2004/11/17 Version 1.4a

General: new option: <code>final</code>	128	when reading files (reported by <i>Michael Hoppe</i>)	88
<code>\MT@cfg@catcodes</code> : fix: reset some more catcodes			

2004/11/26 **Version 1.4b**

General: fix: set catcodes before reading global configuration file (reported by <i>Christoph Bier</i>)	130	form abccz (reported by <i>Georg Verwey</i>)	88
optimisation: use less <code>\expandafers</code> and <code>\csnames</code>	44	<code>\MT@get@slot</code> : don't define <code>\MT@char</code> globally (save stack problem)	91
Protrusion: harmonise dashes in upshape and italic (<code>cmr</code> , <code>pad</code> , <code>pp1</code>)	154	<code>\MT@ifdimen</code> : don't set <code>\MT@count</code> globally (save stack problem)	46
slanted like italics	163	<code>\MT@setup@PDF</code> : new message if <code>\pdfoutput</code> is changed	134
<code>\MT@checklist@family</code> : fix: don't try alias family name if encoding failed	61	<code>\MT@use@set</code> : don't use undeclared font sets	110
<code>\MT@get@basefamily</code> : fix: failed for font names of the			

2004/12/15 **Version 1.5**

General: defaults: step: 4 (suggested by <i>Hàn Thế Thành</i>)	128	<code>\MT@get@highlevel</code> : don't test defaults if called after begin document	106
new option: selected, by default false (suggested by <i>Hàn Thế Thành</i>)	126	<code>\MT@scale@factor</code> : warning for factors outside limits	66
Documentation: add 'Short history'	30	<code>\MT@scale@to@em</code> : don't use <code>\lcode</code> and <code>\rcode</code> for the calculation	64
Inheritance: remove <code>\ss</code> from T1 list, add <code>\DJ</code>	148	<code>\MT@set@ex@codes</code> : allow non-selected font expansion	70
Protrusion: settings for Bitstream Charter	155	<code>\MT@set@pr@codes</code> : adjust protrusion factors before setting the inheriting characters	62
<code>\DeclareMicrotypeAlias</code> : remove spaces around arguments	111	<code>\MT@setup@expansion</code> : defaults: calculate step as $\min(\text{stretch}, \text{shrink})/5$	136
<code>\MT@cfg@catcodes</code> : reset catcode of <code>'=</code> ' (compatibility with Turkish <code>babel</code>)	88	defaults: turn off expansion for DVI output	135
<code>\MT@fix@catcode</code> : reset catcode of <code>'^</code> ' (compatibility with <code>chemsym</code>)	35	disable automatic expansion for DVI output	136

2005/01/24 **Version 1.6**

General: defaults: turn off expansion for old pdfTeX versions	129	tune CMR math letters (OML encoding)	183
load a font if none is selected	56	<code>\MT@get@charwd</code> : use e-TeX's <code>\fontcharwd</code> , if available	65
new option: factor, by default 1000	128	<code>\MT@get@inh@list</code> : correct message if selected is false	91
restructure dtx file	143	<code>\MT@set@ex@codes</code> : introduce factor option	70
test whether <code>\pickup@font</code> has changed	102	<code>\MT@set@pr@codes</code> : introduce factor option	62
test whether numeric options receive a number	128	<code>\MT@setup@expansion</code> : disable automatic expansion for old pdfTeX versions	137
use e-TeX's <code>\ifcsize</code> and <code>\ifdefined</code> if defined	44	<code>\MT@use@set</code> : retain current set if new set is undeclared	110
Protrusion: add italic uppercase Greek letters	163	<code>\MT@vinfo</code> : new macro instead of <code>\ifMT@verbose</code>	36
improve settings for numbers (pointed out by <i>Peter Muthesius</i>)	156		

2005/02/02 **Version 1.6a**

Documentation: add table of fonts with tailored protrusion settings	21	reported by <i>Bernard Gaulle</i>)	91
<code>\MT@get@slot</code> : completely redone, hopefully more robust (compatible with <code>frenchpro</code> ; problem		<code>\MT@pdf@tex@no</code> : new macro	39
		<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older pdfTeX versions	70

2005/03/23 **Version 1.7**

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i>)	107	<code>\textbackslash</code> to T1 encoding	158
disallow automatic expansion if pdfTeX too old	119	<code>\LoadMicrotypeFile</code> : new command (suggested by <i>Andreas Böhmann</i>)	112
fix: remove space after <code>autoexpand</code>	119	<code>\Microtype@Hook</code> : new command for font package authors	130
new value for verbose option: errors	128	<code>\microtypesetup</code> : fix: warning also when setting to (no)compatibility	131
shorter command names	51	<code>\MT@begin@catcodes</code> : also use inside configuration commands	88
warning when running in draft mode	134	<code>\MT@cfg@catcodes</code> : reset catcode of <code>'=</code> ' (compatibility with <code>french*</code> packages)	88
Documentation: add hint about compatibility	26		
remove table of match order (now table 4 on page 89)	12		
Protrusion: fix: remove <code>\</code> from OT1, add			

\MT@DeclareMicrotypeAlias: may also be used inside configuration files	112	\MT@scale: new macro: use e-TeX's \numexpr if available	51
\MT@get@listname@: use \@tfor (<i>Andreas Böhmann's</i> idea)	89	\MT@set@ex@codes: two versions of this macro	70
\MT@get@slot: remove backslash hack	91	\MT@split@name: don't define \MT@encoding &c. globally	60
test for \chardefed commands	92	\MT@test@ast: make it simpler	106
test whether \<encoding>\<...> is defined	92	\MT@try@order: always check for size, too (suggested by <i>Andreas Böhmann</i>)	89
\MT@if@list@exists: don't define \MT@#1@c@name globally, here and elsewhere	90	fix: also check for //<series>/<shape>/ (reported by <i>Andreas Böhmann</i>)	89
\MT@if@dimen: comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i>)	46	\MT@warn@code@too@large: new macro: type out maximum protrusion factor	66
\MT@increment: use e-TeX's \numexpr if available	50	\MT@warn@err: new macro: for verbose=errors	36
\MT@is@composite: new macro: construct command for composite character; no uncontrolled expansion	97	\showhyphens: modify \showhyphens	138

2005/06/23 **Version 1.8**

General: \SetProtrusion: new key: unit	118	\MT@find@file: no longer wrap names in commands	87
if font substitution has occurred, set up the substitute font, not the selected one	100	\MT@fix@fontdimen@six: new macro: test whether \fontdimen 6 is defined	59
new option: config to load a different main configuration file	129	\MT@get@charwd: warning for missing (resp. zero-width) characters	65
new option: unit, by default character	129	\MT@get@listname@: made recursive	89
Documentation: add example for factor option	13	\MT@get@slot: fix: expand active characters	91
add example of how to get rid of a widow (suggested by <i>Adam Kucharczyk</i>)	15	test whether \<encoding>\<...> is defined made more robust	92
add hint about error messages	27	\MT@get@unit: new macro: get unit for codes	67
Font aliases: declare ppr and txr as aliases of ppl resp. ptm	145	\MT@in@rlist: made recursive	49
Font sets: add U encoding to allmath	143	\MT@is@active: new macro: translate inputenc-defined characters	95
Inheritance: remove \DJ from T1 list (it's the same as \DH)	148	\MT@is@letter: warning for non-ASCII characters	94
Protrusion: add LY1 characters for Times	163	\MT@ledmac@setup: character protrusion with ledmac	53
settings for AMS math fonts	187	\MT@map@clist@n: new macro: used instead of \@for	48
verified settings for slanted Computer Modern Roman	171	\MT@map@tlist@n: new macro: used instead of \@tfor	48
\add@accent: fix: disable micro-typographic setup inside \add@accent (reported by <i>Stephan Hennig</i>)	102	\MT@old@cmd: renamed commands from \..MicroType.. to \..Microtype..	36
\DeclareMicrotypeAlias: warning when overriding an alias font	111	\MT@pdf@tex@no: case 5: pdfTeX 1.30	40
\DeclareMicrotypeSetDefault: new command: set default font set	110	\MT@permute@@@@: add ranges to the beginning of the lists	123
\MT@cfg@catcodes: reset catcodes of the remaining ASCII characters	88	\MT@scale: fix: remove spaces in e-TeX variant (reported by <i>Mark Rossi</i>)	51
\MT@check@rlist: made recursive	124	\MT@setupfont@hook: restore % and # when hyperref is loaded	54
\MT@curr@list@name: new macro: current list type and name	98	restore csquotes's active characters	54
\MT@declare@sets: warning when redefining a set	106	restore percent character if Spanish babel is loaded	54
\MT@define@set@key@: use comma lists instead of token lists	106	\MT@split@codes: get character width once only	63
		\MT@use@set: fix: remove braces in first line	110
		\MT@xadd: simplified	47

2005/10/28 **Version 1.9**

General: \DeclareMicrotypeSet: new key: font	109	Documentation: add hint about verbatim environment	25
\SetProtrusion: value 'relative' renamed to 'character' for key unit	118	add remark about Type 1 fonts required for automatic font expansion	8
allow context-specific font setup	100	Font aliases: declare qpl and qtm (qfonts, TeX Gyre) as aliases of ppl resp. ptm	145
compatibility with TeX Live hack (reported by <i>Herrbert Voß</i>)	39	Font sets: add OT4 encoding to text sets	143
disable microtype setup inside hyperref's \pdfstringdef (reported by <i>Hàn Thế Thành</i>)	55	add T5 encoding to text sets	143
fix: use true as the default value	125	Inheritance: add list for OT4	149
option unit: rename value relative to character	129	add list for T5 (requested by <i>Hàn Thế Thành</i>)	150

Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR	158	<code>\MT@get@opt</code> : new key ‘preset’ to set all characters to the specified value before loading the lists	67
settings for OT4 encoding (Computer Modern Roman, Palatino, Times)	154	<code>\MT@is@active</code> : redone: use <code>\set@display@protect</code>	95
settings for T5 encoded Computer Modern Roman	154	<code>\MT@is@letter</code> : using <code>\catcode</code> should be more efficient than inspecting the <code>\meaning</code>	94
<code>\DisableLigatures</code> : new command: disable ligatures (requires pdfTeX 1.30)	112	<code>\MT@maybe@do</code> : redone	60
<code>\microtypecontext</code> : new command: change setup context in the document	103	<code>\MT@rem@from@clist</code> : new macro: remove an item from a comma list	48
<code>\MT@checklist@family</code> : fix: add two missing <code>\expandafters</code>	61	<code>\MT@scale@factor</code> : generalised	66
<code>\MT@detokenize@c</code> : fix the \LaTeX version	45	<code>\MT@setup@expansion</code> : disable expansion if both step and shrink are zero	137
<code>\MT@exp@two@n</code> : new macros: less <code>\expandafters</code>	44	warning if user requested zero step	136
		<code>\MT@toks</code> : use instead of <code>\toks@</code>	42
		<code>\SetProtrusion</code> : (et al.) new key: font	113

2005/12/05 **Version 1.9a**

General: ‘ <code>(file name)/(line number)</code> ’ as default list name	116	diately (requested by <i>Georg Verwey</i>)	106
new option: <code>defersetup</code> , by default true	127	<code>\MT@get@highlevel</code> : no longer check whether defaults have changed	106
remove superfluous test whether <code>\pickup@font</code> has changed	102	<code>\MT@ifdefined@c@T</code> : new macros: true case only	45
Documentation: add explanation for error message in DVl mode	27	<code>\MT@ifint</code> : use <code>\pdfmatch</code> if available	45
add explanation for error message with non-Type 1 fonts	28	<code>\MT@ifstreq</code> : use <code>\pdfstrcmp</code> if available	47
Font aliases: declare <code>mbch</code> (mathdesign) as an alias of <code>bch</code>	146	<code>\MT@in@clist</code> : fix	48
Protrusion: fix: remove ‘_’ from OT1 encoding	159	<code>\MT@info@missing@char</code> : info instead of warning (after <i>Michael Hoppe</i> reported that the ‘fl’ ligature is missing in Palatino SC)	65
settings for T5 encoded Charter	154	<code>\MT@is@feature</code> : new macro: check for pdfTeX feature	51
<code>\microtypesetup</code> : inside the preamble, accepts all package options	131	<code>\MT@map@clist@n</code> : following \LaTeX 3	48
<code>\MT@check@font@cx</code> : optimise context-sensitive setup	102	<code>\MT@permute@</code> : don’t define permutations for unused encodings	122
<code>\MT@define@set@key</code> : don’t expand variables immediately		<code>\MT@rem@from@clist</code> : fix	48
		<code>\MT@setup</code> : defer setup until the end of the preamble	52

2006/01/20 **Version 1.9b**

General: compatibility with listings: sanitise more catcodes (reported by <i>Holger Uhr</i>)	55	add samples of micro-typographic features	4
compatibility with the <code>extendedchar</code> option of the listings package	56	<code>\MT@features</code> : use throughout the package to adjust to beta-ness	51
Documentation: activate expansion in the distributed PDF	1	<code>\MT@ifdimen</code> : use <code>\pdfmatch</code> if available	46
		<code>\MT@warn@code@too@large</code> : fix calculation with present factor	66

2006/02/02 **Version 1.9c**

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verwey</i>)	22	<code>\MT@define@code@key@font</code> : fix: context was ignored	116
Protrusion: settings for URW Garamond	155	<code>\MT@define@code@key@size</code> : fix: embrace <code>\MT@tempsize</code> in <code>\csname</code> (bug introduced in v1.9b)	115

2006/05/05 **Version 1.9d**

Font sets: <code>md*</code> instead of <code>m</code> series in basic sets	143	<code>\DeclareCharacterInheritance</code> : fix: empty context	120
add QX encoding to text sets	143	<code>\MT@detokenize@n</code> : new macro: use <code>\detokenize</code> if available	45
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i>)	149	<code>\MT@get@ex@opt</code> : fix: evaluate preset	71
Protrusion: settings for QX encoding (contributed by <i>Maciej Eder</i>)	161	<code>\MT@get@font@dimen</code> : warning for zero fontdimen	65
settings for Euro symbols (Adobe, ITC, marvosym)	194	<code>\MT@get@opt</code> : optimise: don’t reset when preset option is set	67
tweak AMS settings	187	set list name before presetting	67

<code>\MT@is@active</code> : support for Unicode (inputenc/utf8) . . .	95	mandatory argument	113
<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>tex4ht</code> is loaded (reported by <i>Peter Dyballa</i>)	54	(et al.) split keys of optional and mandatory argument	113
<code>\SetProtrusion</code> : (et al.) optimise: unify keys for			

2006/07/28 **Version 1.9e**

General: fix: default value for <code>activate</code> : <code>true</code> . . .	126	settings for Euler Roman font	190
Documentation: add hint about unknown encodings include LPPL	249	<code>\DeclareCharacterInheritance</code> : new key ‘inputenc’ to set the input encoding	120
Font aliases: declare <code>zeur</code> and <code>zeus</code> (eulervm) as aliases of <code>eur</code> resp. <code>eus</code> (euler)	146	<code>\MT@rem@from@clist</code> : model after <code>\@removeelement</code>	48
Inheritance: adapt to <code>marvosym</code> ’s changed encoding	151	<code>\MT@setup@</code> : empty <code>\MT@setup@</code> after use (compatibility with the <code>combine</code> class)	52
Protrusion: complete settings for Euler Fraktur and Script fonts	194	<code>\pickup@font</code> : no tracing with <code>trace</code> package	101
fix: forgotten comma in <code>mt-mvs.cfg</code> ; adapt to <code>marvosym</code> ’s changed encoding	194	<code>\SetExpansion</code> : new key: <code>inputenc</code>	113
		<code>\SetProtrusion</code> : (et al.) new key: <code>inputenc</code>	113

2006/09/09 **Version 1.9f**

Protrusion: fix: <code>euler-vm</code> did not load euler settings	191	<code>\MT@reset@context</code> : only reset context if it has actually been changed	104
<code>\MT@curr@list@name</code> : fix: <code>\MessageBreak</code> must not be expanded	98	<code>\MT@set@inh@list</code> : fix: forgotten comma in the features list	120
<code>\MT@gdef@n</code> : new macros: global variants	44	<code>\MT@set@named@keys</code> : new macro: set name first, simplify parsing of optional argument	115
<code>\MT@get@inh@list</code> : fix: input encoding must be set after the inheritance list has been parsed	91	<code>\SetProtrusion</code> : (et al.) set catcodes before parsing optional argument	113
<code>\MT@glet</code> : new macro	44		

2007/01/14 **Version 2.0**

General: compatibility with listings: set catcode of backslash to zero (reported by <i>Steven Bath</i>) . . .	56	new: <code>smallcaps</code>	143
compatibility with <code>soul</code> : register <code>\textls</code> and <code>\lststyle</code>	56	<code>\DeclareMicrotypeBabelHook</code> : new command: interaction with <code>babel</code>	113
new option: <code>babel</code> , by default <code>false</code> (language-dependent setup suggested by <i>Ulrich Dirr</i>) . . .	126	<code>\lststyle</code> : fix: font switches don’t pose a problem anymore	79
new option: <code>letterspace</code> , by default <code>100</code>	128	fix: <code>letterspacing</code> commands may be nested	79
new package <code>letterspace</code> : a stripped-down version, containing the <code>letterspacing</code> commands only	1	new command: <code>letterspacing</code>	79
option ‘ <code>babel</code> ’: fix: switch off French <code>babel</code> ’s short-hands properly (reported by <i>Daniel Flipo</i>) . . .	142	totally redone, using the new <code>\letterspacefont</code>	79
option ‘ <code>babel</code> ’: switch off Turkish <code>babel</code> ’s short-hands	142	<code>\MT@declare@sets</code> : fix: empty size list when redefining set	106
option ‘ <code>unit</code> ’, <code>\SetProtrusion</code> : deprecate value ‘ <code>relative</code> ’ completely	118	<code>\MT@is@symbol</code> : made even more robust	96
Documentation: add hint about how to increase <code>font_max</code> and <code>font_mem_size</code>	28	<code>\MT@load@inputenc</code> : sanitise catcodes before loading input encoding (problem with listings)	68
add hint about warning when tracking and expansion is applied to a font	28	<code>\MT@pdftex@no</code> : case 6: <code>pdfTeX 1.40</code>	40
add remark about ‘ <code>draft</code> ’ option disabling microtype (noted by <i>Michalis Miatidis</i>)	9	<code>\MT@setup@noligatures</code> : maybe disable <code>\MT@noligatures</code> after the preamble	141
qualify hint about web documents with regard to older <code>pdfTeX</code> versions	25	<code>\MT@split@name</code> : adjust to possible <code>letterspacing</code>	60
qualify hints about expansion error messages with regard to older <code>pdfTeX</code> versions	27	<code>\SetExtraKerning</code> : new command: additional kerning	114
Font sets: new: <code>footnotesize</code> and <code>scriptsize</code> . . .	143	<code>\SetExtraSpacing</code> : new command: adjustment of interword spacing	114
		<code>\SetTracking</code> : new command: tracking	114
		<code>\textls</code> : new command: <code>letterspacing</code>	84
		starred version: remove spaces around text	84
		<code>\tracingmicrotypeinpdf</code> : new debug method: mark all fonts with PDF annotations	37

2007/01/21 **Version 2.1**

General: compatibility with pinyin: disable microtype in <code>\py@macron</code> (reported by <i>Sven Naumann</i>)	56	<code>\MT@get@ls@basefont</code> : redone: use <code>\pdfmatch</code> to make it bullet-proof	79
fix: letterspace package forgot to load <code>keyval</code>	42	<code>\MT@orig@pickupfont</code> : compatibility with CJK: also check for its definition	100
<code>\slig</code> : new command: protect ligatures in letter-spaced text	79	<code>\textls</code> : fix: use <code>\hmode@bgroup</code>	84

2007/07/14 **Version 2.2**

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i>)	52	<code>\MT@is@composite</code> : more robust: expand exactly once	97
new option: <code>copyfonts</code>	127	<code>\MT@is@symbol</code> : expand once more (for frenchpro)	96
simplify key declarations	116	<code>\MT@lsfont</code> : use <code>\font@name</code> , not <code>\MT@font</code>	75
use catcode trickery for e-TeX test	38	<code>\MT@lua</code> : (basic) support for LuaTeX	41
Documentation: add hint about error message with pdfTeX 1.40	27	<code>\MT@pdftex@no</code> : case 7: pdfTeX 1.40.4	40
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i>)	25	<code>\MT@preset@aux@space</code> : generalised	69
add overview	5	<code>\MT@set@all@pr</code> : (et al.) allow empty values	63
logo transparency and amusement	1	<code>\MT@set@inputenc@</code> : only load inputenc files if necessary	68
Font aliases: declare <code>chr</code> (<code>chmath</code>) as an alias of <code>bch</code> (reported by <i>Geoff Vallis</i>)	146	<code>\MT@set@tr@codes</code> : disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	76
declare <code>fp9x</code> , <code>fp9j</code> (FPL Neu) as aliases of <code>ppl[xj]</code>	145	possibility to customise interword spacing	76
Font sets: default set for tracking: <code>smallcaps</code>	144	<code>\MT@setup@expansion</code> : warning if stretch or shrink aren't multiples of step	137
Inheritance: remove <code>'-'</code> → <code>'127'</code>	148	<code>\MT@setupfont</code> : don't call <code>\@enc@update</code> anymore	57
Protrusion: settings for Bitstream Letter Gothic	155	only add features that are available with the respective pdfTeX	57
Spacing: add sample	195	<code>\MT@setupfont@hook</code> : restore percent character if Galician babel is loaded	54
Tracking: add ligatures that are to be disabled	152	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letterspaced fonts	63
<code>\DeclareMicrotypeVariants</code> : new command	111	<code>\MT@tracking</code> : remember fonts that shouldn't be letterspaced	75
<code>\DisableLigatures</code> : new optional argument: disable selected ligatures only	112	<code>\MT@tracking@</code> : fix: tracking couldn't be re-enabled	75
<code>\slig</code> : always defined	79	<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	140
<code>\MT@checklist@font</code> : fix: construct font name from characteristics	62	<code>\MT@with@babel@and@T</code> : also inspect class options	52
<code>\MT@copy@font</code> : optionally work on copies of fonts	58	<code>\pickup@font</code> : letterspace: setup inside group	101
<code>\MT@get@basefamily</code> : redone, working on font names and suffixes of arbitrary length	88	<code>\SetTracking</code> : new key 'no ligatures' to disable ligatures of letterspaced fonts	114
<code>\MT@get@charwd</code> : subtract letterspacing amount from width	65	new keys 'spacing' and 'outer spacing' to adjust interword spacing (suggested by <i>Steven E. Harris</i>)	114
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	79	third argument may be empty	114
<code>\MT@ifdimen</code> : employ LuaTeX features if available	46	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code>	104
<code>\MT@ifint</code> : employ LuaTeX features if available	45		
<code>\MT@ifstreq</code> : employ LuaTeX features if available	47		
fix: ϵ -TeX version shouldn't use <code>\x</code> and <code>\y</code> (found by <i>Wiebke Petersen</i>)	47		

2007/12/23 **Version 2.3**

General: disable <code>\microtypecontext</code> in <code>hyperref's</code> <code>\pdfstringdef</code>	55	<code>\microtypecontext</code> : made robust (reported by <i>Stephan Hennig</i>)	103
fix: really switch off Turkish shorthands	142	<code>\MT@begin@catcodes</code> : fix: don't disable <code>\KV@sp@def</code>	88
new value for verbose option: <code>silent</code> (suggested by <i>Karl Berry</i>)	128	<code>\MT@define@set@key@font</code> : font: single asterisk means normal font	109
turned some warnings into errors	128	<code>\MT@in@clist</code> : don't use <code>\x</code> (reported by <i>Peter Meier</i>)	48
Documentation: add kerning sample	18	<code>\MT@is@active</code> : support for extended Unicode (inputenc/utf8x resp. ucs) – experimental	95
add letterspacing illustration	17	<code>\MT@noligatures</code> : fix: set evaluation didn't work (bug introduced in v2.2)	85
<code>\do@subst@correction</code> : remember substitute font for all times (reported by <i>Stephan Hennig</i>)	102	<code>\MT@plain</code> : letterspace: support for <code>eplain/miniltx</code>	38
<code>\slig</code> : redone: extract outer kerns from current letterspacing amount	79	<code>\MT@set@curr@os</code> : adjusting spaces made more reliable	77

<code>\MT@set@tr@codes</code> : also adjust tracking if protrusion is not enabled, and even for <code>letterspace</code> (reported by <i>Stephan Hennig</i>)	76	<code>\MT@SetTracking</code> : sanity check for value	114
possibility to customise outer kerning (suggested by <i>Stephan Hennig</i>)	76	<code>\MT@setup@tracking</code> : enable protrusion when tracking is enabled	139
		<code>\MT@tr@outer@l</code> : only change pre outer space if it contains shrink	81

2008/02/29 **Version 2.3a**

General: fix test for <code>soul</code> under plain \TeX	56	too old for extensions	140
Documentation: add hint about <code>babel</code> having to be loaded first	27	<code>\MT@fix@catcode</code> : fix catcodes earlier, and also for the <code>letterspace</code> package	35
add table of available and enabled features	7	<code>\MT@getkey</code> : fix: <code>key=val</code> in class options list	134
mention <code>soulutf8</code>	30	<code>\MT@set@codes</code> : generalised	63
Protrusion: adjust LMR quotation marks again	160	<code>\MT@setupfont@hook</code> : restore percent character if Mexican <code>babel</code> is loaded	54
<code>\MT@error@doesnt@work</code> : error messages if <code>pdfTeX</code> is			

2008/06/04 **Version 2.3b**

<code>\MT@exp@gcs</code> : new macro: reduce save stack size	44	also check for its definition	100
<code>\MT@font@copy</code> : enable font copies also with protrusion contexts (reported by <i>Nathan Rosenblum</i>)	58	<code>\MT@requires@latex</code> : new macro	38
<code>\MT@get@size@</code> : grouping	108	<code>\MT@set@tr@codes</code> : fix: protrusion adjustment only for new fonts (reported by <i>Wolfram Schaalo</i>)	76
<code>\MT@noligatures@</code> : fix: warning messages for unknown slots	86	<code>\MT@tr@outer@l</code> : fix: only in horizontal mode	81
<code>\MT@orig@pickupfont</code> : compatibility with <code>CJKutf8</code> :		make <code>\spaceskip-aware</code> (<code>ragged2e</code>)	81
		<code>\MT@tr@outer@r@</code> : additional test for horizontal mode	82

2008/11/11 **Version 2.3c**

General: $\text{Lua}\TeX$ supported by default	40	coding (reported by <i>Vasile Gaburici</i>)	150
Documentation: add hint about spacing being experimental	26	<code>\MT@detokenize@c</code> : fix: remove last space only (reported by <i>Ulrich Dirr</i>)	45
add hint about partial incompatibility with <code>CJK</code>	27	<code>\MT@tr@outer@r@</code> : additional test for horizontal mode (reported by <i>Sveinung Heggen</i>)	82
Inheritance: add <code>\textcommabelow[STst]</code> to <code>QX</code> en-			

2009/03/27 **Version 2.3d**

General: fix pinyin compatibility check (reported by <i>Silas S. Brown</i>)	56	(reported by <i>Ulrich Dirr</i>)	80
move setup to the very end (for <i>Colin Rourke</i>)	142	<code>\MT@setup@expansion</code> : default step: 1 for <code>pdfTeX</code> versions ≥ 1.40	136
<code>\ifMT@inannot</code> : use <code>pdfTeXcmds</code> for debugging	37	<code>\MT@tr@outer@r@</code> : don't use <code>\x</code> (reported by <i>Ulrich Dirr</i>)	82
<code>\lststyle</code> : disable for $\text{Lua}\TeX$	79	fix: don't adjust in math mode (reported by <i>Christoph Bier</i>)	82
make invalid in math mode	79	fix: don't adjust inside discretionary (reported by <i>Maverick Woo</i>)	82
<code>\microtypesetup</code> : select font after setup	131	<code>\MT@tr@set@okern</code> : allow empty value for outer kerning	84
<code>\MT@check@active@set</code> : warning for missing default sets	130	<code>\textls</code> : make math mode aware	84
<code>\MT@lua</code> : update for $\text{Lua}\TeX$ 0.36	41		
<code>\MT@set@tr@codes</code> : allow zero tracking	75		
<code>\MT@set@tr@zero</code> : fix: allow switching off tracking			

2009/11/09 **Version 2.3e**

Documentation: suggest to patch <code>\@verbatim</code> instead of <code>\verbatim</code>	26	<i>Karl Karlsson</i>)	197
Expansion: settings for T2A encoding (contributed by <i>Karl Karlsson</i>)	153	<code>\MT@fix@fontdimen@six</code> : fix: gobbling settings with tracking failed (reported by <i>Leo</i>)	59
Font sets: <code>sc*</code> instead of <code>sc</code> in <code>smallcaps</code> set	143	<code>\MT@setup@</code> : make space-unaware (requested by <i>Marcin Borkowski</i>)	52
add T2A encoding	143	<code>\MT@tikz@setup</code> : compatibility with <code>tikz</code> (first reported by <i>Christian Stark</i>)	54
Protrusion: settings for T2A encoding (contributed by <i>Karl Karlsson</i>)	160	<code>\MT@tr@outer@r@</code> : fix: set current kerning and spacing again (found by <i>Lars Rönnbäck</i>)	82
Spacing: settings for T2A encoding (contributed by			

2010/01/10 **Version 2.4**

- General: new file `microtype.lua` containing the lua functions (contributed by *Élie Roux*) 43
- Protrusion: settings for T2A encoded Minion (contributed by *Karl Karlsson*) 160

2013/03/13 **Version 2.5**

- General: allow contexts for LuaTeX 117
- disable ‘DVIOutput’ option for XeTeX 127
- fix: check whether ‘`{file}/<line>`’ list name already exists (reported by *Till A. Heilmann*) 116
- letterspacing with LuaTeX 0.62 75
- new files: `microtype-pdftex.def`, `microtype-xetex.def`, `microtype-luatex.def`, containing engine-specific definitions 38
- protrusion with XeTeX 40
- restore `\space` inside listings (reported by *Rolf Dieterich*) 56
- Documentation: add hint about LuaTeX compatibility 27
- add hint about spacing and `ragged2e` 26
- add hint about dtx source code 28
- include `microtype-logo.dtx` and `microtype-lssample.dtx` 221
- Font aliases: declare `lmsy` and `lmm` as aliases of `cmsy` resp. `cmm` (reported by *Jonas Hogstrom*) 144
- declare `zgmx` etc. (`garamondx`) as aliases of `ugm` 146
- declare Latin Modern Roman (OpenType version) as alias of `lmr` when `fontspec` is loaded 144
- declare TeX Gyre Pagella, Asana Math, Palatino LT Std, and Palatino as aliases of Palatino Linotype (OpenType version) 145
- Font sets: add EU1 and EU2 encodings 143
- Inheritance: add rudimentary list for EU1 and EU2 151
- Protrusion: add default lists for EU1 and EU2 159
- improvements to Computer Modern Roman italics (contributed by *Hendrik Vogt*) 163
- Tracking: add EU2 encoding to default list 152
- `\DeclareCharacterInheritance`: allow more than one encoding 120
- `\DeclareMicrotypeAlias`: ignore spaces 111
- `\ifMT@nofamily`: info if settings are not family-specific (suggested by *Hàn Thê Thành*) 62
- `\LoadMicrotypeFile`: remove all spaces in font name 112
- `\lstyle`: fix: ensure to set up math fonts (reported by *RazorXsr*) 79
- `\MT@define@code@key@family`: compatibility with `fontspec`: remove its internal counter (reported by *Till A. Heilmann*) 115
- `\MT@define@code@key@font`: scrub `fontspec` feature count (found by *Meho R*) 116
- `\MT@do@font`: adapt for LuaTeX 50
- adapt for XeTeX 50
- `\MT@get@slot@`: adapt for LuaTeX (requested by *Georg Duffner*) 92
- adapt for XeTeX 92
- `\MT@if@outer@next`: fix: conflict with `amsmath` (reported by *Scott Pakin*) 82
- `\MT@info@missing@char`: fix error message for XeTeX (reported by *Juan Acevedo*) 65
- `\MT@is@charx`: compatibility with unicode 97
- `\MT@ledmac@setup`: fix to work with XeTeX (reported by *Maieul Rouquette*) 53
- `\MT@ls@set@ls`: allow formulas in optional argument to `\textls` (fix by *Heiko Oberdiek*) 84
- `\MT@microtypecontext`: fix: ensure to set up math fonts (reported by *RazorXsr*) 104
- `\MT@register@subst@font`: only register substituted font if it isn’t registered already (reported by *George Gratzner* and *Josep Maria Font*) 102
- `\MT@register@subst@font@cx`: only register if it isn’t registered already 103
- `\MT@scrubfeatures`: compatibility with `fontspec`: remove its internal counter 60
- `\MT@set@all@pr`: fix: remove space (found by *Meho R*) 63
- `\MT@set@pr@codes`: make info about generic settings encoding-specific (reported by *Sebastian Schubert*) 62
- `\MT@setup@spacing`: warning with `ragged2e` (reported by *Steffen Hoffmann*) 139
- `\MT@setupfont`: select font with `fontspec` (found by *Georg Duffner*) 57
- `\MT@setupfont@hook`: restore `\%` and `\#` when `mathastext` is loaded (found by *Seamus Bradley*) 54

2013/05/23 **Version 2.5a**

- General: use `luatexbase` instead of `luatextra` (contributed by *Élie Roux*) 43
- Documentation: add notes on typesetting the documentation 28
- include OpenType configuration files 201
- `\MT@afteraftergroup`: fix: get outer kerning and spacing of nested letterspacing right 78
- `\MT@get@slot@`: adapt to `luaotfload v2.2` (contributed by *Élie Roux*) 93
- `\MT@led@unhbox@line`: simplified 53
- `\MT@ledmac@setup`: support for `eledmac` 53
- `\MT@ls@outer@k`: add marker for tightly nested letterspacing 85
- `\MT@set@tr@codes`: fix: load font for `fontspec` 77
- `\MT@xspace`: fix outer spacing problem with `xspace` (reported by *Dave*) 83

2016/05/01 **Version 2.6**

- General: load `luaotfload` with LuaTeX 43
- redefine `\MT@setupfont@hook` globally for problem with `tikzposter` (reported by *Sam Mason*) 55
- Documentation: add hint about partial incompatibility with `xeCJK` and `luatexja` 27
- missing characters printed with `Charis SIL` 201

suggest to use etoolbox to patch <code>\verbatim</code>	26	<code>\MT@is@xchar</code> : update for fontspec's TU encoding .	97
Font sets: add TU encoding (notified by <i>Will Robertson</i>)	143	<code>\MT@ledmac@setup</code> : support for <code>reledmac</code>	53
add <code>si</code> and <code>scit</code> to <code>smallcaps</code> set (reported by <i>uli</i>)	143	<code>\MT@luatex@no</code> : update for LuaTeX 0.85 (renamed primitives)	41
new: <code>allmath-nott</code> and <code>alltext-nott</code> (suggested by <i>Karl Berry</i>)	143	<code>\MT@noligatures@</code> : use <code>luaotfload</code> function to keep/inhibit ligatures	86
Inheritance: add TU encoding	151	<code>\MT@orig@pickupfont</code> : (in)compatibility with <code>luatexja</code> : disable unknown slots warnings (reported by <i>Max</i>)	100
Protrusion: add TU encoding to lists	159	(in)compatibility with <code>xeCJK</code> : disable unknown slots warnings (reported by <i>HcN</i>)	100
Tracking: add TU encoding to default list	152	compatibility with <code>xeCJK</code> : pretend that CJK wasn't loaded	100
<code>\DeclareMicrotypeSet</code> : ignore spaces	105	<code>\MT@set@tr@codes</code> : use <code>luaotfload</code> 's <code>kernfactor</code> feature if available	76
<code>\DeclareMicrotypeSetDefault</code> : ignore spaces . . .	110	<code>\MT@xspace</code> : fix outer spacing problem with (not only) algorithm (reported by <i>Henning</i> and <i>Ronnie Marks</i>)	83
<code>\DeclareMicrotypeVariants</code> : ignore spaces	111	<code>\UseMicrotypeSet</code> : ignore spaces	110
<code>\lsstyle</code> : fix: ensure to set up math fonts (reported by <i>kleenstar</i>)	79		
<code>\microtypecontext</code> : allow activate shortcut (reported by <i>Karl Berry</i>)	103		
<code>\MT@declare@sets</code> : fix: undefine lists for redefining	106		
<code>\MT@do@font</code> : speed up for LuaTeX	50		
<code>\MT@engine</code> : fix test with LuaTeX 0.85	39		
<code>\MT@get@slot@</code> : fix: could fail with XeTeX (reported by <i>Christopher Schramm</i>)	92		

2016/05/14 **Version 2.6a**

General: fixes for <code>letterspace</code> package with LuaTeX	49	<code>Vofß</code>	50
<code>\MT@do@font</code> : fix <code>lua</code> function (reported by <i>Herbert</i>		<code>\MT@ls@fontspec@font</code> : fix for value of ± 1000 . . .	78

2017/07/07 **Version 2.7**

General: drop <code>luatexbase</code> with recent L ^A T _E X	43	<code>\MT@check@range@</code> : don't warn for override if conflicting list is loaded	124
warning with <code>minimal</code> class	52	<code>\MT@is@composite</code> : compatibility with L ^A T _E X 2017/01/01 (<code>\DeclareUnicodeComposite</code>) (reported by <i>Ulrike Fischer</i> and <i>jcr</i>)	98
Documentation: mention that additional kerning does not work in math mode (discovered by <i>Daniel</i>)	18	<code>\MT@ls@fontspec@font</code> : fix for 'file:{font}' spec (reported by <i>Reinhard Kotucha</i>)	78
Font aliases: declare aliases for <code>newpx</code>	145	<code>\MT@permute@@@@</code> : don't warn for override if conflicting list is loaded	123
declare aliases for <code>newtx</code>	145	<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older LuaTeX versions	70
declare aliases for <code>tempora</code>	145	<code>\MT@setup@expansion</code> : don't disable automatic expansion for DVI output with LuaTeX	136
declare aliases for <code>XCharter</code>	146	<code>\MT@tikz@setup</code> : compatibility with <code>tikz</code> (again) . .	54
declare Latin Modern Roman as alias of <code>lmr</code> with new L ^A T _E X format (reported by <i>Ulrike Fischer</i>)	144	<code>\MT@warn@tracking@DVI</code> : don't warn for letterspacing in DVI mode with LuaTeX	140
Protrusion: automatically choose correct names for Charis SIL small caps (reported by <i>ltcomdata</i>)	214		
<code>\lsstyle</code> : fix: prevent infinite loop with <code>psnfss</code> and <code>exscale</code> packages (reported by <i>user11126</i> , solution by <i>Ulrike Fischer</i>)	79		

2018/01/14 **Version 2.7a**

General: disallow non-automatic expansion with LuaTeX	119	<code>\MT@get@highlevel</code> : test whether <code>\...default</code> is defined	106
<code>\MT@auto</code> : remove 'autoexpand' for LuaTeX 1.0.6 (reported by <i>Ulrike Fischer</i>)	136	<code>\MT@get@slot</code> : expand active characters earlier . . .	91
with LuaTeX, non-automatic font expansion is no longer possible (as confirmed by <i>Hans Hagen</i>)	136	<code>\MT@info@nottracking@</code> : defer 'No tracking' message	61
		<code>\MT@is@active</code> : compatibility with <code>newunicodechar</code> (reported by <i>Nils Anders Danielsson</i>)	95

2019/02/28 **Version 2.7b**

General: update <code>lua</code> function <code>microtype.info</code> after changes in <code>luaotfload</code> (reported by <i>Moritz Wemheuer</i> and <i>Ulrike Fischer</i>)	43	(reported by <i>Franz Wexler</i>)	149
Documentation: update hint about non-7-bit characters (notified by <i>Frank Mittelbach</i>)	26	<code>\MT@info@missing@char</code> : fix message for glyphs specified as names in XeTeX (reported by <i>Paolo Ney</i>)	65
Inheritance: add <code>textquotedblleft</code> ligature to OT4		<code>\MT@setupfont</code> : always select current font with XeTeX and LuaTeX (reported by <i>Paolo Ney</i> , solution by <i>Ulrike Fischer</i>)	57

2019/10/10 **Version 2.7c**

General: turn warning into info when overwriting the keepligature function (reported by <i>Andy N</i>) ..	86	<code>\MT@is@symbol</code> : take care of <code>\remove@tlig</code>	96
<code>\MT@is@active</code> : compatibility with L ^A T _E X 2019/10/01	95	<code>\showhyphens</code> : compatibility with L ^A T _E X 2019/10/01 (reported by <i>Phelype Oleinik</i> and <i>Falk Hanisch</i>)	138

2019/11/18 **Version 2.7d**

<code>\MT@copy@font@</code> : in LuaT _E X, don't use the <code>\copyfont</code> primitive, but load the font anew (reported by <i>Paolo Polesana</i> and <i>Oliver Kopp</i>)	58	from list (reported by <i>Markus Kohm</i>)	102
<code>\MT@register@subst@font</code> : remove substitute font		<code>\MT@register@subst@font@cx</code> : remove substitute font from lists	103

2020/12/07 **Version 2.8**

General: letterspace works with e-T _E X only	38	<code>\lsstyle</code> : fix: enforce math setup, again	79
compatibility with soul: patch for font change (reported by <i>Md Ayquassar</i>)	56	<code>\microtypecontext</code> : fix activate shortcut	103
fix for luatexbase	43	ignore spaces	103
Documentation: declare DVIoutput option deprec- ated	9	<code>\MT@do@font</code> : fix for X _Y L ^A T _E X	50
squash fake news about automatic font expansion with dvilualatex	8	simplify lua function	50
Font aliases: declare aliases for step and domitian (notified by <i>Daniel Benjamin Miller</i>)	145	<code>\MT@fix@fontdimen@six</code> : try to fix zero <code>\fontdimen 6</code>	59
declare aliases for stix and stix2 fonts	145	<code>\MT@if@luaotf@font</code> : use lua function	49
declare New Computer Modern as an alias of Latin Modern Roman	145	<code>\MT@ifstreq</code> : use X _Y L ^A T _E X's <code>\stringcmp</code>	47
Font sets: default set for expansion: alltext-nott	144	<code>\MT@setup@expansion</code> : warning when expanding in DVI mode with LuaT _E X (reported by <i>Daniel Ben- jamin Miller</i>)	135
default set for spacing: alltext-nott	144	<code>\MT@tr@set@space@</code> : simplified	81
		<code>\MT@tr@unit@</code> : fix: allow unit regardless whether letterspacing is set	78
		<code>\textmicrotypecontext</code> : ignore spaces	104

2021/02/22 **Version 2.8a**

General: rename <code>mt-pad.cfg</code> to <code>mt-EBGaramond.cfg</code> (requested by <i>Karl Berry</i>)	143	Inheritance: specify 'ff' ligature as Unicode instead of glyph name	201
rename <code>mt-PalatinoLinotype.cfg</code> to <code>mt-Palatino.cfg</code> (requested by <i>Karl Berry</i>) ..	201	Protrusion: hide <code>euoritc</code> settings (requested by <i>Karl Berry</i>)	194
Documentation: reorganise table 1 for clarification (suggested by <i>Daniel Benjamin Miller</i>)	7	Spacing: add dummy settings (reported by <i>DORpapt</i>)	198
Font aliases: add -LF, -TLF, -OsF and -T0sF as vari- ants	144	<code>\DeclareCharacterInheritance</code> : skip settings for tracking	120
declare mmodern fonts as aliases of Latin Modern (reported by <i>Daniel Benjamin Miller</i>)	144	<code>\MT@get@size@@</code> : compatibility with svjour3 (repor- ted by <i>Ekkehart Schlicht</i> (no relation!))	108
reference New Computer Modern also by file name (reported by <i>Canageek</i>)	145	<code>\MT@gl@et</code> : use LuaT _E X's <code>\gl@et</code> , if available	44

2021/02/25 **Version 2.8b**

Inheritance: dummy settings for the Font Awesome font (<code>mt-FontAwesome.cfg</code>)	205	settings for the Lato font (<code>mt-Lato.cfg</code>) (reported by <i>dsedivec</i>)	205
----------------------------------------------------------------------------------------------------	-----	-------------------------------------------------------------------------------------------------	-----

2021/03/14 **Version 2.8c**

<code>\ifMT@fontspec</code> : add to hook for fontspec	54	<code>\MT@remove@tlig</code> : fix for text commands containing conditionals (reported by <i>Frank Mittelbach</i>) ..	97
<code>\MT@is@opt@char</code> : fix for optionally defined glyphs (reported by <i>Frank Mittelbach</i>)	96		

D Index

Numbers in upright shape refer to the page where the corresponding entry is described (bold face) resp. occurs. Numbers in *italics* refer to the code line where the corresponding entry is defined (underlined) resp. used.

Options	DVIoutput	9	letterspace	8
	activate	6	protrusion	6
	auto	7	selected	8
	babel	9	shrink	8
	config	9	spacing	6
	draft	9	step	8
	expansion	6	stretch	8
	factor	7	tracking	6
	final	9	unit	7
	kerning	6	verbose	9
Commands	\DeclareCharacterInheritance	20	\SetExtraSpacing	19
	\DeclareMicrotypeAlias	21	\SetProtrusion	13
	\DeclareMicrotypeBabelHook	23	\SetTracking	15
	\DeclareMicrotypeSet*	10	\UseMicrotypeSet	12
	\DeclareMicrotypeSet	10	\slig	24
	\DeclareMicrotypeSetDefault	12	\sstyle	23
	\DeclareMicrotypeVariants*	20	\microtypecontext	22
	\DeclareMicrotypeVariants	20	\microtypesetup	9
	\DisableLigatures	24	\textls*	24
	\LoadMicrotypeFile	22	\textls	23
	\SetExpansion	14	\textmicrotypecontext	22
	\SetExtraKerning	18		
A	a0poster (package)	108	algorithm (package)	239
	activate (option)	6, 126, 235	amsmath (package)	238
	\add@accent	2961	amssymb (package)	187
	\adjustspacing	196, 4365	article (package)	22
	ae (package)	21, 144	auto (option)	7, 8, 126
B	babel (option)	9, 23, 27, 32, 126, 235		
	babel (package)	2, 3, 5, 17, 23, 24, 27,	54, 88, 113, 141, 142, 146, 232, 233, 235–237	
C	chemsym (package)	232	config (option)	9, 20, 32, 129, 233
	chmath (package)	21, 146, 236	contour (package)	134
	CJK (package)	27, 56, 100, 236, 237, 239	copyfonts (option)	58, 126, 127, 236
	CJKutf8 (package)	100, 237	crop (package)	134
	cm-super (package)	8	csquotes (package)	29, 54, 158, 233
	color (package)	134	\curr@fontshape	2905, 2907, 2909, 2916, 2959
	combine (package)	142, 235		
D	\DeclareCharacterInheritance	20, 41, 3802	\define@newfont	2898, 2906, 2910, 2920
	\DeclareMicrotypeAlias	21, 43, 71, 3394	\DisableLigatures	24, 40, 62, 3433
	\DeclareMicrotypeBabelHook	23, 45, 64, 3460	\do@subst@correction	2958
	\DeclareMicrotypeSet	10, 32, 57, 59, 72, 3093	docstrip (package)	34
	\DeclareMicrotypeSet*	10, 3093	domitian (package)	21, 145, 240
	\DeclareMicrotypeSetDefault	12, 34, 61, 3349	draft (option)	9, 126, 128, 235
	\DeclareMicrotypeVariants	20, 42, 58, 63, 3379	dsfont (package)	59
	\DeclareMicrotypeVariants*	20	DVIoutput (option)	9, 126, 127, 238, 240
	defersetup (option)	54, 126, 127, 131, 234		
E	eco (package)	21, 144	euler (package)	98, 190, 235
	\efcode	1557, 1583, 1584, 1625, 1627	eulervm (package)	21, 146, 191, 235
	eledmac (package)	53, 238	euroitc (package)	194, 240
	eplain (package)	24, 31, 34, 38, 52, 134, 236	europs (package)	194
	e-TeX (engine)	33, 38, 40, 45,	eurosans (package)	194
		50, 51, 64, 65, 82, 108, 221, 232–234, 236, 240	\expandglyphsinfont	1539
	etoolbox (package)	26, 239	expansion (option)	6, 11, 125, 126, 135
	euca1 (package)	192	exscale (package)	239
	eufrak (package)	194		

- F**
- \f@family 2912, 2913
 - \f@size 1931, 2905, 2907, 2909, 2916, 2959
 - factor (option) 7, 13, 33, 128, 232
 - fancyvrb (package) 26
 - final (option) 9, 33, 126, 128, 231
 - fix-cm (package) 59
 - \font 972, 1048, 1833, 1981
 - \fontname 111, 1020, 1022, 1024, 1026, 1027, 1037, 1038, 1823, 1831, 1835, 1839, 1877, 1883, 1888, 1891, 1966, 1981, 1993, 1995, 2000, 2002, 2005, 2008, 2011, 2046, 2097, 2234, 2236, 2898, 2906, 2915, 2920, 2940, 2942, 2944, 2960, 2974, 2976, 2996, 3001
- G**
- garamondx (package) 21, 146, 238
 - german (package) 35, 231
 - \glb@currsiz 1969, 3035
- H**
- hfoldsty (package) 21, 144
- I**
- IEEEtran (package) 100
 - \iffontchar 731, 1331, 2755, 2765, 2767, 2834, 2846, 2847
 - \ifMT@auto 304, 4569, 4630
 - \ifMT@babel 304, 4716, 4844
 - \ifMT@do 1087, 1154, 1791, 2281
 - \ifMT@document 345, 3921
 - \ifMT@draft 304, 4462
 - \ifMT@expansion 304, 4518, 4532, 4627, 4673
 - \ifMT@fontspec 850, 1067, 3164, 3584, 3610
 - \ifMT@if@ 303, 869, 876, 899, 931, 2992, 4028, 4864, 4867
 - \ifMT@inannot 99
 - \ifMT@inlist@ 624, 669, 777, 807, 992, 1121, 1138, 1147, 1165, 1183, 1789, 2385, 2396, 2445, 2494, 2975, 2986, 2998, 3063, 3075, 3416
 - \ifMT@kerning 304, 4730, 4857
- J**
- jurabib (package) 100
- K**
- kerning (option) 6, 11, 31, 126
 - keyval (package) 42, 63, 147, 236
 - \knaccode 1740, 1741, 1751, 1754, 1760
- L**
- ledmac (package) 29, 32, 53, 100, 233
 - ledpar (package) 29
 - \leftmarginkern 820
 - letterspace (option) 8, 17, 24, 32, 45, 75, 128, 235
 - letterspace (package) 1, 24, 31, 32, 34, 36, 38, 54, 76, 77, 79, 235–237, 239, 240
 - \letterspacefont 1839
 - lineno (package) 53
 - listings (package) 26, 55, 56, 68, 88, 234, 235, 238
 - lmodern (package) 8, 160
 - \LoadMicrotypeFile 22, 44, 74, 3412, 4467
 - \lpcode 1222, 1258, 1259, 1463, 1466, 1866
- M**
- marvosym (package) 21, 32, 34, 151, 194, 234, 235
 - mathastext (package) 54, 238
 - mathdesign (package) 21, 146, 234
 - memoir (package) 23, 100
 - \Microtype@Hook 130, 4296
 - \microtypecontext 22, 47, 920, 3024, 3043, 4469, 4827, 4830
 - microtypecontext (environment) 22
 - \microtypesetup 9, 46, 4300, 4468
 - MiKTeX (distribution) 28
 - miniltx (package) 24, 31, 34, 38, 52, 236
 - minimal (package) 52, 56, 239
 - \fontcharwd 1285, 1289, 1295
 - \fontdimen 1052, 1053, 1057, 1058, 1061, 1320, 1325, 1425, 1428, 1852, 1853, 1883, 1993, 2039, 2040, 2047, 2048, 2055, 2057, 2063, 2078, 2097, 2247, 2250
 - fontinst (package) 198, 221, 223
 - fontinstallationguide (package) 177
 - \fontname 1831, 1835
 - fontspec (package) 6, 25, 27, 30, 31, 43, 60, 78, 97, 107, 115, 144, 151, 238–240
 - fourier (package) 59
 - french (package) 232
 - frenchpro (package) 96, 232, 236
 - \glb@settings 1970
 - graphics (package) 134
 - hyperref (package) 54, 55, 100, 134, 233, 236
 - \ifMT@nofamily 1193, 1200
 - \ifMT@noligatures 304, 4810
 - \ifMT@nonselected 1519, 2507, 2515
 - \ifMT@norest 2576, 2593, 2606, 2620, 2681
 - \ifMT@opt@auto 4039, 4573, 4605
 - \ifMT@opt@DVI 4039, 4487
 - \ifMT@opt@expansion 4039, 4516, 4674
 - \ifMT@protrusion 304, 814, 837, 4499, 4689
 - \ifMT@selected 304, 4613, 4633
 - \ifMT@spacing 304, 4698, 4715
 - \ifMT@tracking 304, 4685, 4745
 - \ifMT@xunicode 850, 2781, 2838
 - ifpdf (package) 100, 134
 - \iftracingmicrotypeinpdfall 105
 - \ignoreligaturesinfont 2285
 - inputenc (package) 14, 27, 32, 56, 68, 91, 95, 98, 233, 235, 236
 - \knbcode 1733, 1734, 1750, 1753, 1759
 - \knbscode 1655, 1656, 1679, 1683, 1689
 - \lslig 24, 54, 1986, 1989
 - \lststyle 23, 51, 922, 946, 1967, 2173, 2179, 2184, 2187, 2197, 2201, 4470
 - luaotfload (package) 27, 30, 43, 50, 70, 76, 86, 93, 238, 239
 - LuaTeX (engine) 1, 4, 6–9, 12–16, 20, 23–25, 27, 29–31, 34, 37–41, 43–45, 50, 53, 57, 58, 70, 75–77, 83, 85, 86, 92, 117, 127, 133, 135, 136, 140, 144, 147, 151, 159, 236–240
 - luatexbase (package) 43, 238–240
 - luatexja (package) 27, 100, 238, 239
 - luatextra (package) 238
 - MinionPro (package) 107
 - mlmodern (package) 21, 144, 240
 - \MT@font 406, 995, 1018, 1057, 1059, 1321, 1335, 1803–1805, 1807, 1810, 2448, 2520, 2945
 - \MT@abbr@ex 753
 - \MT@abbr@ex@c 753
 - \MT@abbr@ex@inh 753
 - \MT@abbr@kn 753
 - \MT@abbr@kn@c 753
 - \MT@abbr@kn@inh 753
 - \MT@abbr@nl 753
 - \MT@abbr@pr 753

- `\MT@abbr@pr@c` [753](#)
`\MT@abbr@pr@inh` [753](#)
`\MT@abbr@sp` [753](#)
`\MT@abbr@sp@c` [753](#)
`\MT@abbr@sp@inh` [753](#)
`\MT@abbr@tr` [753](#)
`\MT@abbr@tr@c` [753](#)
`\MT@active@features`
 [1029](#), [2980](#), [2983](#), [2995](#), [3008](#), [3052](#), [3062](#),
 [3441](#), [4500](#), [4628](#), [4686](#), [4699](#), [4731](#), [4819](#), [4820](#)
`\MT@addto@annot` [99](#)
`\MT@addto@setup` [798](#),
 [799](#), [880](#), [1438](#), [2889](#), [3025](#), [3026](#), [3671](#),
 [3689](#), [4153](#), [4301](#), [4461](#), [4776](#), [4818](#), [4843](#), [4876](#)
`\MT@adjustspacing` [4361](#), [4362](#), [4629](#)
`\MT@afteraftergroup` [1895](#), [1903](#), [1912](#)
`\MT@auto` [1529](#), [4568](#), [4583](#), [4593](#), [4596](#), [4600](#)
`\MT@auto@` [1529](#), [1552](#), [1597](#)
`\MT@autofalse` [306](#), [4581](#), [4592](#)
`\MT@autotrue` [306](#), [4255](#), [4258](#), [4608](#)
`\MT@babelfalse` [313](#)
`\MT@babeltrue` [313](#)
`\MT@begin@catcodes`
 [2386](#), [2387](#), [2427](#), [3024](#), [3028](#), [3041](#),
 [3094](#), [3319](#), [3350](#), [3381](#), [3395](#), [3420](#), [3436](#),
 [3470](#), [3486](#), [3508](#), [3529](#), [3543](#), [3808](#), [4271](#), [4272](#)
`\MT@cat` [1434](#), [2880](#), [2882](#)
`\MT@cfg@catcodes` [933](#), [1456](#), [2412](#), [2429](#)
`\MT@char` [1247](#), [1248](#), [1258](#), [1259](#),
 [1264](#), [1265](#), [1268](#), [1270](#), [1285](#), [1286](#), [1289](#),
 [1463](#)–[1467](#), [1583](#), [1584](#), [1586](#), [1587](#), [1625](#)–
 [1627](#), [1655](#), [1656](#), [1662](#), [1663](#), [1669](#), [1670](#),
 [1673](#), [1674](#), [1679](#)–[1684](#), [1733](#), [1734](#), [1740](#),
 [1741](#), [1744](#), [1745](#), [1750](#)–[1754](#), [2293](#), [2294](#),
 [2296](#), [2549](#), [2565](#), [2573](#), [2575](#), [2585](#), [2587](#),
 [2595](#), [2596](#), [2601](#), [2605](#), [2606](#), [2611](#), [2613](#),
 [2619](#), [2621](#), [2624](#), [2628](#), [2632](#), [2636](#), [2752](#),
 [2755](#), [2757](#), [2759](#), [2761](#), [2825](#), [2829](#), [2834](#),
 [2836](#), [2839](#), [2841](#), [2847](#), [3860](#)–[3862](#), [3867](#)–[3869](#)
`\MT@char@` [1291](#), [1292](#), [1295](#), [1330](#),
 [1331](#), [2549](#), [2556](#), [2560](#), [2565](#), [2614](#), [2685](#),
 [2687](#), [2693](#), [2694](#), [2696](#), [2709](#), [2710](#), [2713](#),
 [2714](#), [2717](#), [2718](#), [2722](#), [2724](#), [2758](#), [2760](#),
 [2788](#), [2791](#), [2795](#), [2801](#), [2808](#), [2810](#), [2837](#), [2854](#)
`\MT@charstring` [2562](#), [2757](#), [2773](#), [2800](#)
`\MT@charxstring` [2803](#), [2841](#)
`\MT@check@active@set`
 [4282](#), [4506](#), [4646](#), [4688](#), [4708](#), [4735](#)
`\MT@check@font` [991](#), [2971](#), [3054](#)
`\MT@check@font@cx` [2981](#), [3054](#)
`\MT@check@range` [3999](#), [4001](#)
`\MT@check@range@` [4001](#), [4002](#)
`\MT@check@rlist` [3948](#), [3992](#)
`\MT@check@rlist@` [3992](#), [3993](#)
`\MT@check@step` [4634](#)
`\MT@checklist@` [1095](#), [1114](#), [2278](#)
`\MT@checklist@family` [1132](#)
`\MT@checklist@font` [1176](#)
`\MT@checklist@size` [1160](#)
`\MT@checksetup` [4312](#), [4323](#), [4330](#), [4349](#), [4385](#)
`\MT@clear@options` [163](#), [296](#), [788](#), [4244](#), [4454](#)
`\MT@clist@break` [596](#), [1127](#), [1155](#), [1171](#), [1185](#)
`\MT@clist@function` [596](#)
`\MT@cnt@encoding` [3887](#), [3895](#), [3896](#)
`\MT@cnt@family` [3893](#), [3902](#), [3903](#)
`\MT@cnt@series` [3900](#), [3909](#), [3910](#)
`\MT@cnt@shape` [3907](#), [3915](#), [3916](#)
`\MT@config@file` [4260](#), [4269](#), [4270](#), [4274](#), [4275](#), [4278](#)
`\MT@context` [2470](#), [2482](#), [2501](#), [2528](#)
`\MT@copy@font` [978](#),
 [1014](#), [3667](#), [3673](#), [3685](#), [3692](#), [4162](#), [4164](#), [4490](#)
`\MT@copy@font@` [1014](#), [3667](#), [3673](#), [3685](#), [3692](#), [4162](#)
`\MT@count` [512](#), [513](#),
 [737](#), [1277](#), [1285](#), [1287](#), [1289](#), [1293](#), [1295](#),
 [1298](#), [1304](#), [1309](#), [1310](#), [1314](#), [1325](#), [1357](#), [1419](#)
`\MT@curr@file` [2390](#), [2391](#), [2400](#), [2401](#),
 [3422](#), [3423](#), [3641](#), [3643](#), [3646](#), [3832](#), [4274](#), [4878](#)
`\MT@curr@list@name` ... [1323](#), [1360](#), [1450](#), [1593](#),
 [2290](#), [2371](#), [2539](#), [2849](#), [2857](#), [2862](#), [2870](#), [2876](#)
`\MT@curr@ls` [1874](#), [1990](#), [1993](#)
`\MT@curr@set@name` [3118](#), [3120](#)–
 [3122](#), [3124](#), [3127](#), [3128](#), [3133](#), [3138](#), [3142](#),
 [3143](#), [3152](#), [3177](#), [3181](#), [3210](#), [3238](#), [3261](#),
 [3267](#), [3271](#), [3272](#), [3523](#), [3599](#), [3620](#), [3635](#),
 [3654](#), [3726](#), [3730](#), [3737](#), [3740](#), [3742](#), [3748](#),
 [3751](#), [3754](#), [3766](#), [3780](#), [3789](#), [3797](#), [3833](#), [3835](#)
`\MT@declare@char@inh` [3814](#), [3822](#), [3830](#)
`\MT@declare@sets` [3101](#), [3106](#), [3119](#), [3443](#)
`\MT@DeclareMicrotypeAlias` [3396](#), [3398](#)
`\MT@DeclareMicrotypeSetDefault` [3351](#), [3353](#)
`\MT@DeclareSet` [3097](#), [3099](#), [3115](#)
`\MT@DeclareSetAndUseIt` [3096](#), [3114](#)
`\MT@DeclareVariants` [3383](#), [3384](#), [3386](#)
`\MT@def@bool@opt` [4116](#),
 [4129](#), [4130](#), [4134](#), [4148](#), [4160](#), [4170](#), [4182](#)
`\MT@def@n` [393](#), [3089](#), [3090](#)
`\MT@default@ex@set` [3368](#)
`\MT@default@kn@set` [3368](#)
`\MT@default@pr@set` [3368](#)
`\MT@default@sp@set` [3368](#)
`\MT@default@tr@set` [3368](#)
`\MT@define@code@key`
 ... [3567](#), [3627](#), [3629](#), [3630](#), [3842](#), [3844](#), [3845](#)
`\MT@define@code@key@family` [3578](#), [3628](#), [3843](#)
`\MT@define@code@key@font` [3604](#), [3632](#), [3847](#)
`\MT@define@code@key@size` [3592](#), [3631](#), [3846](#)
`\MT@define@opt@key` .. [3633](#), [3656](#)–[3659](#), [3793](#)–[3795](#)
`\MT@define@optionX` [4300](#), [4358](#), [4361](#)
`\MT@define@optionX@` [4375](#), [4404](#), [4406](#), [4407](#)
`\MT@define@set@key@` [3131](#), [3311](#)–[3314](#)
`\MT@define@set@key@font` [3259](#), [3316](#)
`\MT@define@set@key@size` [3170](#), [3315](#)
`\MT@detokenize@c` [436](#), [2557](#), [2753](#), [2769](#)
`\MT@detokenize@n` [436](#), [2831](#)
`\MT@dimen@six`
 ... [1050](#), [1278](#), [1314](#), [1357](#), [2072](#), [2074](#), [2243](#)
`\MT@dinfo` [87](#)
`\MT@dinfo@list`
 [1112](#), [1122](#), [1125](#), [1130](#), [1139](#), [1142](#), [1148](#),
 [1150](#), [1158](#), [1166](#), [1169](#), [1174](#), [1184](#), [1187](#), [1191](#)
`\MT@dinfo@nl` [87](#)
`\MT@DisableLigatures` [3433](#)
`\MT@do@font` [704](#), [1224](#), [1557](#), [1692](#), [1761](#), [1866](#)
`\MT@documentfalse` [345](#)
`\MT@documenttrue` [345](#), [4822](#)
`\MT@dofalse` ... [1087](#), [1099](#), [1126](#), [1143](#), [1170](#), [1188](#)
`\MT@dofont@function` [708](#), [719](#)
`\MT@dotrue` [1087](#), [1090](#), [1123](#), [1140](#), [1149](#), [1167](#), [2273](#)
`\MT@draftfalse` [309](#), [4184](#)
`\MT@drafttrue` [309](#), [4186](#)

- \MT@edef@n 395, 2506, 2532, 3079,
 3573, 3587, 3643, 3646, 3649, 3831, 3979, 4216
 \MT@encoding 1064, 1180, 1201, 1203, 1206, 2465,
 2477, 2557, 2753, 2769, 2830, 2861, 2868, 2876
 \MT@end@catcodes 2388, 2431, 3025, 3032,
 3042, 3112, 3335, 3366, 3392, 3410, 3428,
 3445, 3481, 3505, 3524, 3540, 3554, 3825, 4273
 \MT@endinput 167, 4239, 4248
 \MT@engine 175,
 277, 281, 287, 293, 2174, 2178, 4240, 4246
 \MT@engine@tooold 175, 278
 \MT@error 76, 780, 883,
 2174, 2185, 2363, 2376, 3208, 3344, 3373,
 3451, 3676, 3702, 3709, 4172, 4335, 4353,
 4393, 4574, 4586, 4606, 4675, 4746, 4759, 4839
 \MT@error@doesnt@work 4742
 \MT@ex@cc@name 1511,
 1513, 1599, 1600, 1611, 1617, 1618, 1630, 3489
 \MT@ex@context 1018, 3046, 3088
 \MT@ex@doc@contexts 3088
 \MT@ex@factor 314, 1530, 1603
 \MT@ex@factor@ . 1530, 1559, 1564, 1573, 1574, 1597
 \MT@ex@inh@name 1585–1587
 \MT@ex@level 314, 4629, 4631
 \MT@ex@max 326, 1576, 1577
 \MT@ex@min 326, 1579, 1580
 \MT@ex@setname 3337
 \MT@ex@split@val 1571
 \MT@exp@cs 390, 393, 396, 399, 404, 1030,
 1032, 1041, 1164, 1269, 1344, 1347, 1377,
 1381, 1391, 1394, 1397, 1587, 1674, 1745,
 1915–1917, 2493, 2999, 3009, 3010, 3014,
 3065, 3076, 3176, 3598, 3618, 3869, 3950, 3997
 \MT@exp@gcs 390, 394, 398, 400, 405
 \MT@exp@one@n 407, 441, 603, 982,
 984, 1042, 1118, 1181, 1788, 2395, 2971,
 2974, 2984, 2996, 3002, 3020, 3062, 3073,
 3088, 3106, 3310, 3329, 3360, 3415, 3822, 4827
 \MT@exp@two@c 409, 427, 432, 442, 638, 981, 1022,
 1024, 1026, 1027, 1036, 1834, 1966, 2002,
 2005, 2008, 2554, 2561, 2756, 2824, 2839, 2840
 \MT@exp@two@n 411, 1136, 1145, 3137, 3266
 \MT@expandfont 1515, 1533, 1537
 \MT@expansion 998, 1502, 4666
 \MT@expansionfalse 305, 4524, 4529, 4623
 \MT@expansiontrue 305, 4254
 \MT@extra@context 3088, 3473, 3489, 3512,
 3532, 3546, 3613, 3616, 3617, 3619, 3660,
 3668, 3674, 3686, 3691, 3804, 3947, 3951,
 3954, 3957, 3958, 3963, 3968, 3969, 3971, 3998
 \MT@extra@inputenc 3802, 3829, 3834, 3835
 \MT@factor@default 338, 4223, 4503
 \MT@family 982, 1064,
 1137, 1180, 1201, 1203, 1206, 2466, 3406, 3407
 \MT@familyalias
 983, 984, 1076, 1144, 1146, 2476, 2478, 3408
 \MT@feat 1087, 1250, 1339, 1341, 1343,
 1344, 1346, 1347, 1353, 1355, 1358, 1366–
 1370, 1372, 1374–1378, 1381, 1382, 1387,
 1391, 1394, 1397, 1400–1402, 1411, 1412,
 1417, 1435, 1809, 1971, 2361, 2363, 2366,
 2367, 2371, 2373, 2376, 2502, 2504, 2506,
 2510, 2514, 2518, 2519, 2522, 2530, 2532,
 2533, 2535, 2542, 2546, 2850, 2851, 2880, 2882
 \MT@features
 773, 3088, 3101, 3310, 3324, 3355, 3638, 3813
 \MT@features@long 773, 776, 781, 3059, 3827
 \MT@file@list 2382,
 2384, 2392, 2395, 2397, 2402, 2405, 3415, 3419
 \MT@find@file 982, 984, 2382
 \MT@fix@catcode 5
 \MT@fix@font@set 3160, 4474
 \MT@fix@fontdimen@six 1050, 1075
 \MT@font 406, 731, 733,
 975, 981, 1036, 1043, 1052–1055, 1058,
 1061, 1222, 1223, 1258, 1259, 1264, 1265,
 1285, 1286, 1289, 1292, 1295, 1320, 1325,
 1331, 1425, 1428, 1463, 1464, 1466, 1467,
 1542, 1546, 1552, 1557, 1583, 1584, 1625,
 1627, 1655, 1656, 1662, 1663, 1669, 1670,
 1679–1681, 1683, 1684, 1689–1691, 1733,
 1734, 1740, 1741, 1750, 1751, 1753, 1754,
 1759, 1760, 1788, 1792, 1872, 1873, 2282,
 2572, 2627, 2884, 2940–2942, 2944, 2959,
 2960, 2971, 2972, 2977, 2984, 3002, 3012, 3021
 \MT@font@copy . 1018, 1026, 1028, 1036–1038, 1048
 \MT@font@list
 .. 2884, 2971, 2972, 2974, 2976, 2977, 3053
 \MT@font@orig 1020
 \MT@font@sets 3141, 3160, 3270, 4474
 \MT@fontspecfalse 850
 \MT@fontspectrue 850, 851, 855, 893
 \MT@gdef@n 393, 3464, 3480, 3504, 3539,
 3553, 3742, 3766, 3780, 3837, 3880–3885, 4290
 \MT@get@axis 3284–3287, 3297
 \MT@get@basefamily 2394, 2432
 \MT@get@basefamily@ 2438, 2441
 \MT@get@char@unit 1249, 1389, 1415, 1507
 \MT@get@charwd 1283, 1314, 1392, 1415
 \MT@get@config 4260
 \MT@get@ex@opt 1506, 1524, 1597
 \MT@get@ex@opt@ 1605–1607, 1609, 1616
 \MT@get@font 3265, 3275
 \MT@get@font@ 3276, 3281, 3625
 \MT@get@font@and@size 3609, 3624
 \MT@get@font@dimen 1318, 1395
 \MT@get@highlevel ... 3136, 3146, 3299, 3572, 3583
 \MT@get@inh@list 1211, 1509, 1642, 1720, 2528
 \MT@get@listname 2447, 2504, 2530
 \MT@get@listname@ 2447
 \MT@get@ls@basefont 1827, 1992, 1999
 \MT@get@opt 1209, 1364, 1640, 1718
 \MT@get@range 3174, 3184, 3596
 \MT@get@size 3191, 3196, 3205, 3224, 3295
 \MT@get@size@ 3231, 3242
 \MT@get@size@@ 3242
 \MT@get@slot 1246, 2292, 2549, 3859, 3866
 \MT@get@slot@ 2566, 2570
 \MT@get@space@unit
 .. 1389, 1491, 1653, 1660, 1667, 1731, 1738
 \MT@get@tr@opt 1814, 1934
 \MT@get@tr@opt@ 1954–1957, 1959
 \MT@get@unit 1397, 1405, 1943
 \MT@get@unit@ 1405
 \MT@getkey 4441, 4456
 \MT@gl@et . 387, 400, 894, 1022, 1916, 2002, 2033,
 2179, 2187, 2938, 3035, 3054–3057, 3408,
 3667, 3685, 3890, 4162, 4164, 4790, 4796, 4882

- \MT@glet@nc 399, 1059, 1206, 1810, 2046, 2234, 2542, 2960, 3053, 3124, 3127, 3133, 3261, 3497, 3726, 3730, 3737, 3740, 3748, 3751, 3754, 3789
 \MT@glet@nn 404, 3072, 4287
 \MT@if@false 303, 865, 872, 895, 912, 2989, 3996, 4858, 4865
 \MT@if@list@exists 1199, 1505, 1523, 1639, 1717, 1813, 2501
 \MT@if@luaotf@font 687, 707, 1829, 1864, 1873, 2295, 2308
 \MT@if@outer@next 2113, 2128, 2132, 2135, 2140, 2145–2148
 \MT@if@true 303, 866–868, 873–875, 896–898, 925, 929, 930, 2982, 4005, 4009, 4017, 4022, 4859–4863, 4866
 \MT@if@defined@c@T 415, 869, 901, 1267, 1303, 1585, 1672, 1743, 1858, 1948, 2740, 3162, 3406, 3834, 3937, 4296, 4299, 4885
 \MT@if@defined@c@TF 415, 815, 1817, 1842, 1845, 1856, 1990, 2069, 2240, 2287, 2288, 2505, 2531, 3946
 \MT@if@defined@n@T 415, 804, 1115, 1133, 1161, 1177, 1268, 1400, 1436, 1586, 1611, 1673, 1744, 1938, 1946, 1960, 2484, 2879, 3121, 3401, 3492, 3650, 3896, 3903, 3910, 3916, 3947, 3957, 4441
 \MT@if@defined@n@TF 415, 1076, 1093, 1116, 1134, 1162, 1178, 1201, 1366, 1374, 1599, 1617, 1804, 1914, 2276, 2366, 2472, 2557, 3149, 3338, 3341, 3369, 3641, 3923, 3978, 3987, 4057, 4283, 4286, 4647, 4825
 \MT@if@dim 535, 658, 659, 663, 664, 3207, 3216, 4003–4005, 4007, 4008, 4015–4017, 4020, 4021
 \MT@if@dimen 497, 3233, 3729, 3739, 3753, 4232
 \MT@if@empty 448, 1222, 1223, 1256, 1262, 1480, 1481, 1652, 1659, 1666, 1689–1691, 1703–1705, 1707–1709, 1730, 1737, 1759, 1760, 1776, 1777, 2052, 2053, 2061, 2077, 2209, 2232, 2233, 2245, 3064, 3100, 3104, 3148, 3158, 3185, 3186, 3199, 3200, 3289, 3300, 3323, 3327, 3354, 3358, 3440, 3519, 3634, 3640, 3660, 3666, 3674, 3684, 3691, 3764, 3812, 3819, 4070, 4101, 4263, 4309, 4382
 \MT@if@fint 461, 2716, 3520, 3765, 4215, 4225
 \MT@if@streq 544, 1105, 1442, 1609, 2362, 2503, 2518, 2789, 2792, 2800, 2809, 3019, 3264, 3608, 3728, 3750, 3752, 3814, 3821, 3938, 3958, 4029, 4073, 4075, 4078, 4081, 4103, 4105, 4119, 4120, 4192, 4193, 4198, 4202, 4231, 4311, 4318, 4322, 4329, 4384, 4390, 4718
 \MT@in@clist 624, 776, 805, 1118, 1136, 1145, 1181, 1788, 2384, 2395, 2971, 2974, 2984, 2996, 3062, 3415
 \MT@in@rlist 652, 1164, 2493
 \MT@in@rlist@ 652
 \MT@in@rlist@ 652
 \MT@in@tlist 640, 2444, 3073
 \MT@in@tlist@ 640
 \MT@in@not@false 100
 \MT@in@not@true 100
 \MT@increment 737, 3895, 3902, 3909, 3915
 \MT@info 76, 91, 96, 1057, 4199, 4425, 4429
 \MT@info@missing@char 1298, 1328, 2623
 \MT@info@nl 76, 92, 93, 97, 816, 1202, 1329, 4190, 4200, 4270, 4284, 4288, 4486, 4490, 4502, 4509, 4630, 4667, 4687, 4692, 4701, 4711, 4734, 4738, 4751, 4766, 4835, 4846
 \MT@info@not@tracking 996, 1106, 1110
 \MT@info@not@tracking@ 1106, 1110
 \MT@inh@do 2541, 3848
 \MT@inh@feat 3802, 3812, 3817, 3828
 \MT@inh@split 3850, 3854
 \MT@inlist@false 624, 628, 641, 653, 2992
 \MT@inlist@true 624, 630, 648, 660, 665, 2992
 \MT@is@active 2554, 2729
 \MT@is@char 2561, 2756, 2773
 \MT@is@charx 2782, 2803, 2840
 \MT@is@composite 2559, 2827
 \MT@is@feature 775, 3105, 3328, 3359
 \MT@is@letter 2555, 2673, 2761, 2836
 \MT@is@number 2699, 2704, 2824
 \MT@is@opt@char 2755, 2765
 \MT@is@symbol 2558, 2751
 \MT@is@tlig 2759, 2817
 \MT@is@uni@comp 2834, 2846
 \MT@is@xchar 2793, 2799
 \MT@iterate 674
 \MT@kerning 1002, 1714, 4737
 \MT@kerning@false 311
 \MT@kerning@true 311
 \MT@kn@c@name 1722, 1724, 1767, 3546
 \MT@kn@context 3047, 3088
 \MT@kn@doc@contexts 3088
 \MT@kn@factor 314
 \MT@kn@factor@ 1366
 \MT@kn@inh@name 1743–1745
 \MT@kn@max 326
 \MT@kn@min 326
 \MT@kn@setname 3337
 \MT@kn@split@val 1728
 \MT@kn@unit 320
 \MT@kn@unit@ 1374, 1770
 \MT@led@unhbox@line 816
 \MT@led@mac@setup 810, 889–891
 \MT@let@cn 401, 404, 405, 1077, 1215, 1513, 1600, 1646, 1724, 1939, 1947, 2361, 2373, 2498, 2502, 2535, 2940, 3228, 3654, 3833
 \MT@let@nc 399, 1402, 2154, 2514, 2546, 2987, 3980, 3985, 4079, 4082, 4325, 4332, 4749, 4764
 \MT@let@nn 404, 1367, 1372, 1375, 1387, 1618, 1621, 1961
 \MT@letterspace 341, 1817, 1947, 1949, 1951, 4799, 4800, 4802
 \MT@letterspace@ 1233, 1303, 1304, 1817, 1818, 1821, 1824, 1832, 1839, 1844, 1852, 1865–1867, 1874, 1883, 1927–1930, 2074, 2077, 2195, 2210–2213, 2246, 2249
 \MT@letterspace@default 341, 4800
 \MT@listname 2447, 2495, 2498, 2505, 2506, 2508, 2510, 2531, 2532, 2534, 2535, 2539, 2542, 3869, 3874
 \MT@listname@count 3637, 3642, 3644
 \MT@load@input@enc 1443, 1446, 1455
 \MT@load@list 1213, 1511, 1644, 1722, 2358
 \MT@loop 674, 682, 730, 3251, 3984
 \MT@lower 3178, 3184, 3599
 \MT@ls@adjust 1881, 2206
 \MT@ls@adjust@ 2190, 2214

- `\MT@ls@adjust@empty` 2192, 2206
`\MT@ls@adjust@relax` 2193, 2206
`\MT@ls@basefont` 1999, 2008, 2011, 2012
`\MT@ls@fontspec@colon` 1923
`\MT@ls@fontspec@font` 1834, 1923
`\MT@ls@outer@k` 1884,
 1889, 1897, 2090, 2104, 2151, 2169, 2254
`\MT@ls@set@ls` 2197, 2200, 2206
`\MT@ls@too@large` 2213, 2216, 4802
`\MT@lsfont` 1823,
 1833, 1839, 1852, 1853, 1862, 1866, 1867,
 1872, 1966, 1977, 2022, 2024, 2039, 2040,
 2047, 2048, 2055, 2057, 2063, 2078, 2247, 2250
`\MT@ltx@pickupfont` 918, 959, 964, 2955, 2963
`\MT@lua` 192, 255, 356,
 364, 485, 522, 568, 688, 709, 2296, 2309, 2577
`\MT@lua@copy@font` 1047
`\MT@lua@copyfont` 1027, 1047
`\MT@luatex@no` 253, 354
`\MT@map@clist@` 596
`\MT@map@clist@c` 596,
 1029, 2291, 2983, 2995, 3008, 3052, 3059,
 3101, 3324, 3355, 3638, 3813, 3817, 3827, 4439
`\MT@map@clist@n` 596, 1092, 2275, 3088, 3103, 3123, 3134,
 3172, 3262, 3310, 3326, 3357, 3387, 3462,
 3570, 3581, 3594, 3606, 3746, 3762, 3863,
 4065, 4068, 4097, 4099, 4128, 4210, 4307, 4380
`\MT@map@tlist@` 614
`\MT@map@tlist@c` 614, 643, 654, 1032, 1269,
 1587, 1674, 1745, 2451, 3014, 3948, 3997, 4474
`\MT@map@tlist@n` 614, 2423, 2775, 3889
`\MT@max@char` 2673, 2696
`\MT@max@slot` 2673, 2722
`\MT@maybe@do` .. 1087, 1196, 1503, 1637, 1715, 1790
`\MT@maybe@gobble@with@tikz` 856, 1913, 1970
`\MT@maybe@rem@from@list` 3014, 3018
`\MT@microtype@context` 3024
`\MT@minus` 1832, 1927–1930
`\MT@MT` 2, 76, 79, 82, 84,
 86, 91, 92, 94, 95, 292, 787, 884, 886, 3676,
 4242, 4264, 4425, 4429, 4487, 4722, 4839, 4840
`\MT@MT@pickupfont` 961, 966, 2955, 2965
`\MT@next@listname` 2454, 2463
`\MT@next@listname@` 2473, 2485, 2491
`\MT@nl@ligatures` 2282, 3433
`\MT@nl@setname` 2274, 3433
`\MT@nofamily@false` 1193, 1198
`\MT@nofamily@true` 1193, 2464
`\MT@noligatures` 1006, 2270, 4811
`\MT@noligatures@` 2022, 2024, 2282, 2285
`\MT@noligatures@false` 308
`\MT@noligatures@true` 308, 3442
`\MT@nonselected@false` 1519, 1534
`\MT@nonselected@true` 1519, 1522
`\MT@noreset@false`
 .. 2682, 2689, 2698, 2719, 2789, 2801, 2809
`\MT@noreset@true` 2552, 2682, 2707
`\MT@old@cmd` 66, 71–74
`\MT@opt@autofalse` 4040
`\MT@opt@autotrue` 4040, 4130
`\MT@opt@def@set` 4056, 4084, 4109
`\MT@opt@DVIfalse` 4041
`\MT@opt@DVIftrue` 4041, 4137, 4140
`\MT@opt@expansion@false` 4039
`\MT@opt@expansion@true` 4039
`\MT@options` 3557
`\MT@optwarn@admissible` 4042, 4121, 4202
`\MT@optwarn@nan` 4046, 4217, 4227
`\MT@orig@add@accent` 2961
`\MT@orig@foreign@language` 4852, 4854
`\MT@orig@pickupfont` 2890, 2957
`\MT@orig@py@macron` 956, 960, 965
`\MT@orig@select@language` 4847, 4849
`\MT@outer@kern` 1883, 1887, 1889, 1890, 1894, 1902,
 1993, 1994, 1996, 2151, 2152, 2169, 2227, 2266
`\MT@outer@space` 1877,
 1878, 2037, 2086, 2088, 2089, 2102, 2103,
 2120, 2121, 2136, 2137, 2141, 2142, 2167
`\MT@pdf@annot` 99
`\MT@pdf@or@lua` 180, 188, 290, 464, 687, 4783
`\MT@pdf@tex@no` 212, 353
`\MT@permute` 3479, 3503, 3517, 3538, 3552, 3840, 3879
`\MT@permute@` 3879
`\MT@permute@@` 3879
`\MT@permute@@@` 3879
`\MT@permute@@@@` 3879
`\MT@permute@@@@@` 3914, 3919
`\MT@permute@@@@@@` 3927, 3929
`\MT@permute@define` 3920, 3930–3932, 3976
`\MT@permute@reset` 3889, 3982
`\MT@permutelist` 3473, 3489,
 3515, 3532, 3546, 3613, 3614, 3619, 3838,
 3942, 3943, 3947, 3951, 3954, 3957–3959,
 3961–3963, 3971, 3972, 3998, 4030, 4032, 4033
`\MT@pickupfont` 2955
`\MT@plain` 125, 146
`\MT@pr@c@name` 1213, 1215, 1471, 3473
`\MT@pr@context` 1018, 3046, 3088
`\MT@pr@doc@contexts` 3088
`\MT@pr@efactor` 314, 4226, 4503, 4504
`\MT@pr@efactor@` 1366
`\MT@pr@inh@name` 1267, 1268, 1270
`\MT@pr@level` 314, 4501, 4502
`\MT@pr@max` 326
`\MT@pr@min` 326
`\MT@pr@setname` 3337
`\MT@pr@split@val` 1254
`\MT@pr@unit` 320, 4233, 4505
`\MT@pr@unit@` 1374, 1474
`\MT@preset@aux` 1476,
 1478, 1480, 1481, 1484, 1772, 1774, 1776, 1777
`\MT@preset@aux@factor` 1476, 1484, 1703–1705, 1772
`\MT@preset@aux@space` . 1478, 1484, 1707–1709, 1774
`\MT@preset@ex` 1612, 1629
`\MT@preset@kn` 1765
`\MT@preset@kn@` 1765
`\MT@preset@pr` 1469
`\MT@preset@pr@` 1469
`\MT@preset@sp` 1696
`\MT@preset@sp@` 1696
`\MT@ProcessOptionsWithKV` 4433, 4457
`\MT@protrude@chars` 4358, 4362, 4501, 4689
`\MT@protrusion` 997, 1195, 4508
`\MT@protrusion@false` 304
`\MT@protrusion@true` 304, 4249
`\MT@rbba@expansion` 768
`\MT@rbba@kerning` 768
`\MT@rbba@protrusion` 768
`\MT@rbba@spacing` 768

- `\MT@rbba@tracking` 768
`\MT@register@font` 1010, 2972, 3055
`\MT@register@font@cx` 3007, 3055
`\MT@register@subst@font` 2946, 2973, 3056
`\MT@register@subst@font@cx` 2994, 3056
`\MT@rem@from@clist` 635, 1042, 2977, 3002, 3020, 3562
`\MT@rem@from@list` 1032, 1040
`\MT@rem@last@space` 436
`\MT@remove@tlig` 2820, 2823
`\MT@repeat` 674, 684, 733, 3254, 3990
`\MT@requires@latex` 145, 164, 796, 914, 1965, 2888, 4048, 4209, 4434, 4438, 4881
`\MT@requires@luatex` 348, 387, 988, 1055, 1231, 1538, 1539, 1561, 1608, 1784, 2285, 3779, 3785, 4133, 4363, 4377, 4519, 4596, 4604, 4683
`\MT@requires@pdftex` 348, 463, 498, 545, 811, 977, 987, 1001, 1005, 1016, 1230, 1275, 1301, 1560, 1636, 1714, 1783, 2019, 2271, 3434, 3664, 3778, 3783, 4159, 4253, 4376, 4540, 4571, 4599, 4682, 4809
`\MT@res@a` ... 558, 560, 626, 633, 636, 638, 642, 647
`\MT@res@b` 559, 560, 636–638, 646, 647
`\MT@reset@context` 3034, 3038, 3044, 3071
`\MT@reset@context@` 3044, 3071
`\MT@reset@ef@codes` 1508, 1532, 1560, 1613
`\MT@reset@ef@codes@` 1555, 1565, 1569
`\MT@reset@kn@codes` 1719, 1726, 1756
`\MT@reset@kn@codes@` 1756
`\MT@reset@pr@codes` 1210, 1217, 1226
`\MT@reset@pr@codes@` 1226
`\MT@reset@sp@codes` 1641, 1648, 1686
`\MT@reset@sp@codes@` 1686
`\MT@restore@catcodes` 5, 7, 8, 159, 165, 4887
`\MT@restore@p@h` 847, 876, 931
`\MT@saved@setupfont` 4422, 4430
`\MT@scale` 744, 1278, 1340, 1354, 1357, 1574
`\MT@scale@factor` ... 1280, 1311, 1338, 1486, 1631
`\MT@scale@to@em` .. 1257, 1263, 1275, 1308, 1492, 1654, 1661, 1668, 1732, 1739, 1950, 2070, 2241
`\MT@scrubfeature` 1068, 1080, 3585
`\MT@scrubfeatures` 1080, 3165, 3611
`\MT@selectedfalse` 307
`\MT@selectedtrue` 307
`\MT@series` 1064, 1180, 2467, 2479
`\MT@set@all@ex` 1555, 1632
`\MT@set@all@kn` 1756, 1778
`\MT@set@all@pr` 1219, 1226, 1482
`\MT@set@all@sp` 1686, 1711
`\MT@set@babel@context` 4824, 4850, 4855, 4869
`\MT@set@codes` . 1216, 1237, 1514, 1647, 1725, 2374
`\MT@set@curr@ls` 1874
`\MT@set@curr@ok` 1894, 1896, 1902, 1905, 2125, 2129
`\MT@set@curr@os` 1876, 1904, 2125, 2129
`\MT@set@default@set` 3355, 3360, 3368
`\MT@set@ex@codes` 1536, 4614, 4616
`\MT@set@ex@codes@n` 1519, 1536, 4616
`\MT@set@ex@codes@s` 1504, 4614
`\MT@set@ex@heirs` 1587, 1624
`\MT@set@inh@list` 3809, 3811
`\MT@set@inputenc` 1212, 1433, 1510, 1643, 1721, 2540
`\MT@set@inputenc@` 1436, 1438
`\MT@set@kn@codes` 1716
`\MT@set@kn@heirs` 1745, 1749
`\MT@set@listname` 1214, 1365, 1512, 1598, 1645, 1723, 1936, 2849
`\MT@set@lsbasefont` 2008
`\MT@set@lsfont` 1871, 1964, 1977
`\MT@set@named@keys` 3475, 3491, 3513, 3534, 3548, 3557
`\MT@set@pr@codes` 1197
`\MT@set@pr@heirs` 1271, 1461
`\MT@set@sp@codes` 1638
`\MT@set@sp@heirs` 1674, 1678
`\MT@set@tr@codes` 1800, 1972, 1982
`\MT@set@tr@zero` 1819, 2008, 4404
`\MT@SetExpansion` 3487, 3489
`\MT@SetExtraKerning` 3544, 3546
`\MT@SetExtraSpacing` 3530, 3532
`\MT@SetProtrusion` 3471, 3473
`\MT@SetTracking` 3509, 3511
`\MT@setup@` 794, 798, 4151, 4152, 4882
`\MT@setup@contexts` 3033, 3051
`\MT@setup@copies` 4473, 4489
`\MT@setup@expansion` 4476, 4513
`\MT@setup@kerning` 4480, 4729, 4772
`\MT@setup@noligatures` 4481, 4807
`\MT@setup@PDF` 4472, 4484, 4494
`\MT@setup@protrusion` 4475, 4497
`\MT@setup@spacing` 4479, 4696, 4773
`\MT@setup@spacing@check` 4714, 4885, 4886
`\MT@setup@tracking` 4477, 4681, 4744, 4771
`\MT@setup@wartracking` 4478, 4776
`\MT@setupfont` .. 973, 2949, 4423, 4426, 4430, 4466
`\MT@setupfont@hook` 864, 894, 900, 905, 931, 938, 976
`\MT@shape` 1064, 1180, 2468, 2480
`\MT@shorthandoff` 4833, 4864, 4867
`\MT@show@pdfannot` 106
`\MT@shrink` 323, 1527, 4536, 4537, 4544, 4545, 4548, 4552, 4619, 4632
`\MT@shrink@` 1527, 1542, 1546, 1552, 1597
`\MT@shrink@default` 339
`\MT@size` 659, 663, 664, 1056, 1064, 1180
`\MT@size@name` 652, 2495
`\MT@SOUL@doword` 949, 950
`\MT@sp@cname` 1644, 1646, 1698, 3532
`\MT@sp@context` 3047, 3088, 4718
`\MT@sp@doc@contexts` 3088
`\MT@sp@factor` 314
`\MT@sp@factor@` 1366
`\MT@sp@inh@name` 1672–1674
`\MT@sp@max` 326
`\MT@sp@min` 326
`\MT@sp@setname` 3337
`\MT@sp@split@val` 1650
`\MT@sp@unit` 320
`\MT@sp@unit@` 1374, 1701
`\MT@spacing` 1002, 1635, 4710
`\MT@spacingfalse` 310
`\MT@spacingtrue` 310
`\MT@split@codes` 1239, 1243
`\MT@split@name` 981, 1064
`\MT@step` 323, 1528, 4539, 4541, 4559, 4563, 4566, 4633, 4636, 4637, 4641
`\MT@step@` 1528, 1542, 1546, 1552, 1597
`\MT@stretch` 323, 1526, 4533, 4534, 4537, 4544, 4546, 4551, 4554, 4618, 4632
`\MT@stretch@` 1526, 1542, 1546, 1552, 1597
`\MT@stretch@default` 339, 4534
`\MT@strip@prefix` 2782, 2803, 2839
`\MT@temp` ... 1221–1224, 1688–1692, 1758–1761, 2042, 2046, 2049, 2057, 2060, 2117, 2120,

- 2126, 2130, 2133, 2136, 2141, 2149, 2151,
2157, 2231, 2234, 2251, 3283, 3308, 4437,
4442, 4447, 4448, 4450, 4453, 4648, 4652, 4656
\MT@tempencoding 3922, 3923, 3933, 3939
\MT@tempfamily 3934
\MT@tempseries 3935
\MT@tempshape 3936
\MT@tempsize 3890, 3937, 3946, 3948, 3952
\MT@test@ast 2068, 2239, 3147, 3156
\MT@text@microtypecontext 3041
\MT@textls 49, 2192, 2193, 2195
\MT@textmicrotypecontext 3041
\MT@the@pr@code 1228, 1258, 1264, 1868
\MT@the@pr@code@tr 1228, 1868
\MT@tikz@setup 856, 877, 954
\MT@tlist@break 614, 649, 671, 2456, 4036
\MT@toks 302,
1329, 1360, 1593, 2553, 2588, 2615, 2687,
2694, 2744, 2795, 2810, 2819, 2854, 2866, 2874
\MT@tr@cc@name 1938,
1939, 1946, 1947, 1960, 1961, 3514, 3521
\MT@tr@context 3047, 3088
\MT@tr@doc@contexts 3088
\MT@tr@factor@ 1937
\MT@tr@font@list 1786
\MT@tr@ispace 1842, 1843, 1954
\MT@tr@ligatures 1858, 1957, 2021, 2024
\MT@tr@max 326, 2217–2219
\MT@tr@min 326, 2221–2223
\MT@tr@noligatures 1858, 2018
\MT@tr@okern 1856, 1956
\MT@tr@ospace 1845, 1846, 1954
\MT@tr@outer@icr 2130, 2159
\MT@tr@outer@icr@ 2159
\MT@tr@outer@l 1879, 2082
\MT@tr@outer@next 2110, 2114, 2161
\MT@tr@outer@r 1906, 2110, 2126, 2133, 2159
\MT@tr@outer@r@ 2111, 2116
\MT@tr@set@okern 1857, 2227
\MT@tr@set@okern@ 2232, 2233, 2238
\MT@tr@set@ospace 1848, 2038
\MT@tr@set@ospace@ 2043–2045, 2051
\MT@tr@set@ospace@@ 2054, 2059, 2062, 2067
\MT@tr@setname 3337
\MT@tr@unit@ 1938, 2069, 2240
\MT@tracking 989, 1786, 1972, 2951, 4404, 4405, 4691
\MT@tracking@ 1786, 4405
\MT@trackingfalse 312, 4748
\MT@trackingtrue 312
\MT@try@order 2451, 2459
\MT@undefined@char 2734, 2739, 2750
\MT@upper 3178, 3184, 3599
\MT@use@set 3324, 3329, 3337
\MT@UseMicrotypeSet 3320, 3322
\MT@val 3064–3066, 3068, 3074, 3076,
3079, 3135, 3139, 3147, 3150, 3153, 3173–
3175, 3187, 3190, 3192, 3195, 3197, 3198,
3204, 3206, 3207, 3209, 3212, 3214, 3225,
3228, 3233–3235, 3237, 3239, 3244, 3263–
3265, 3268, 3277–3279, 3291, 3293, 3298,
3300, 3302, 3304, 3571, 3573, 3582, 3585,
3587, 3595–3597, 3607–3609, 3615, 3620,
3862, 3869, 3873, 3874, 4057, 4058, 4061,
4069, 4070, 4073, 4075, 4078, 4081, 4100,
4101, 4103, 4105, 4308, 4309, 4311, 4318,
4322, 4329, 4335, 4381, 4382, 4384, 4390, 4393
\MT@variants 2444, 3379, 3390
\MT@vinfo 76, 93, 995, 1107, 1111, 1369,
1378, 1382, 1417, 1601, 1619, 1803, 1821,
2299, 2310, 2367, 2391, 2401, 2404, 2508,
2510, 2516, 3045, 3068, 3417, 3423, 4190,
4197, 4314, 4320, 4326, 4333, 4387, 4392, 4826
\MT@warn@ascii 2696, 2853
\MT@warn@axis@empty 3290, 3301, 3306
\MT@warn@code@too@large 1344, 1347, 1351
\MT@warn@err 76, 4194, 4195
\MT@warn@ex@too@large 1577, 1580, 1591
\MT@warn@maybe@inputenc ... 2289, 2867, 2875, 2878
\MT@warn@nodim 3718, 3731, 3741, 3755
\MT@warn@number@too@large 2723, 2859
\MT@warn@preset@towidth ... 1475, 1495, 1702, 1771
\MT@warn@rest 2594, 2631, 2864
\MT@warn@tracking@DVI 1822, 4778
\MT@warn@unknown ... 2584, 2600, 2610, 2635, 2872
\MT@warn@unknown@once 2890, 2891
\MT@warning . 67, 76, 94, 2218, 2222, 2302, 2519,
3122, 3151, 3237, 3307, 3402, 3425, 3522,
3651, 3720, 3768, 3781, 3786, 3940, 3961,
4032, 4194, 4199, 4296, 4373, 4412, 4414, 4415
\MT@warning@nl 76, 88, 95, 158, 279, 786, 790, 826,
838, 907, 1321, 1358, 1410, 1450, 1496,
1592, 1806, 2030, 2854, 2860, 2865, 2873,
2922, 3494, 3693, 4043, 4050, 4061, 4143,
4195, 4200, 4234, 4241, 4277, 4291, 4463,
4520, 4564, 4620, 4639, 4703, 4719, 4784, 4871
\MT@while@num 680, 704, 710, 934–936
\MT@with@babel@and@T
... 803, 866–868, 896–898, 4859–4863, 4866
\MT@with@package@T 800, 851, 853,
870, 873–875, 877, 882, 889–893, 903, 915,
929, 930, 932, 954, 955, 2890, 2891, 2933, 4702
\MT@xadd 104, 580, 2392, 2397, 2402,
2405, 2999, 3010, 3076, 3176, 3419, 3598, 3869
\MT@xaddb 588, 3618, 3950
\MT@xdef@n . 395, 3339, 3342, 3371, 3376, 3405,
3521, 3613, 3635, 3797, 3835, 3971, 4058, 4060
\MT@xetex@no 244
\MT@xspace 2149, 2164
\MT@xspace@ 2164
\MT@xunicodefalse 852
\MT@xunicodetrue 852, 853, 892
- N** newcomputermodern (package) 145
newpx (package) 21, 59, 145, 239
newtx (package) 21, 59, 145, 239
- O** \outputmode 194, 4133
- newunicodechar (package) 95, 239
\normalfont 972, 4659, 4663
\nullfont 972

- P**
- \pdfadjustinterwordglue 4406, 4700
 - \pdfadjustspacing 196, 950, 4365, 4369, 4661
 - \pdfappendkern 4408, 4733
 - \pdfcopyfont 1026
 - pdfcpot (package) 26, 30, 54, 64, 99, 231
 - \pdffeedback 1055
 - \pdffontexpand 1539, 1542, 1546, 1552
 - \pdffontsize 1054, 1055
 - \pdfnoligatures 2285, 2301, 2307
 - \pdfprependkern 4407, 4732
 - \pdfprotrudechars 195, 4364, 4368, 4795
 - pdfTeX (engine) . 1, 4, 6–9, 13–19, 23–35, 37–39, 41–43, 45, 46, 53, 57–59, 64, 70, 72, 73,
 - 75, 76, 79, 80, 83, 85, 112, 115, 117, 119, 129, 133, 134, 136, 137, 140, 197, 199, 231–237
 - pdftexcmds (package) 37, 237
 - \pdftracingfonts 358
 - \pickup@font 2921, 2923, 2932, 2955–2957
 - pifont (package) 100
 - pinyin (package) 56, 236, 237
 - \protrudechars 195, 4364
 - protrusion (option) 6, 11, 125, 126
 - psnfss (package) 79, 198, 239
 - pstricks (package) 134
 - pxfonts (package) 21, 145
- Q**
- qfonts (package) 21, 145, 233
- R**
- ragged2e (package) 26, 81, 139, 237, 238
 - reledmac (package) 53, 239
 - resize (package) 108
- S**
- selected (option) 8, 14, 33, 69, 90, 126, 232
 - \selectfont 1973, 3037, 3049, 4301, 4877
 - \set@fontsize 3244
 - \SetExpansion 14, 36, 3484
 - \SetExtraKerning 18, 38, 3542
 - \SetExtraSpacing 19, 39, 3527
 - \SetProtrusion 13, 35, 3468
 - \SetTracking 15, 37, 3507
 - \sfcode 4717
 - shapepar (package) 29, 53
 - \shbscode 1669, 1670, 1681, 1684, 1691
 - \showhyphens 4647
 - shrink (option) .. 8, 15, 33, 128, 136, 138, 234, 236
- T**
- \tagcode 1862, 2288, 2294
 - tempora (package) 21, 145, 239
 - tex4ht (package) 54, 235
 - TeX Live (distribution) 28, 39, 146, 233
 - \textls 23, 52, 53, 923, 947, 1882, 1968, 1985, 2190
 - \textls* 24
 - \textmicrotypecontext 22, 48, 919, 3041
 - tikz (package) 54, 227, 237, 239
 - tikzposter (package) 238
- U**
- ucs (package) 95, 236
 - ulgothic (package) 21
 - \UnicodeEncodingName 2832
 - unit (option) 7, 32, 233, 235
- V**
- verbose (option) 9, 31, 32, 128, 232, 233, 236
- W**
- wordcount (package) 52, 236
- X**
- XCharter (package) 21, 146, 239
 - xeCJK (package) 27, 100, 238, 239
 - X_YTeX (engine) 1, 4, 6, 7, 9, 12, 13, 20, 29, 31, 34, 38, 40, 50, 53, 57, 59, 83, 93, 133, 138, 140, 144, 147, 151, 159, 238–240
 - \XeTeXcharglyph 2621
 - \XeTeXfonttype 2572
 - \XeTeXglyph 1292
 - \XeTeXglyphindex 2608
 - \XeTeXglyphname 2627
 - \XeTeXlastfontchar 733
 - \XeTeXprotrudechars 4372
 - \xspace 2148, 2149
 - xspace (package) 83, 238
 - \xspaceskip 2096, 2099
 - xunicode (package) 97, 98, 238
 - xunicode-addon (package) 100
- Y**
- yfonts (package) 24
- Z**
- zefonts (package) 21, 144

E The L^AT_EX Project Public License

LPPL Version 1.3c 2008-05-04

Copyright 1999, 2002–2008 L^AT_EX3 Project

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

Preamble

The L^AT_EX Project Public License (LPPL) is the primary license under which 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.

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

- 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.
- 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.
- 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.
- 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 Main-

tainer 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
%   https://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.