

The microtype package

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

R Schlicht
w.m.l@gmx.net

v2.3e
2009/11/09

Abstract

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

Note that font expansion and character protrusion will only work with pdf \TeX , at least version 0.14f. Automatic font expansion requires version 1.20 or newer. Disabling ligatures requires pdf \TeX 1.30, letterspacing and the adjustment of interword spacing and of kerning requires version 1.40. The package will by default enable protrusion and expansion if they can safely be assumed to work. These two features are also available with lua \TeX . The `microtype` package does not work with Xe \TeX .

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 © 2004–2009 R Schlicht. It may be distributed and/or modified under the conditions of the [\$\LaTeX\$ Project Public License](#), either version 1.3c of this license or (at your option) any later version. This work has the LPPL maintenance status ‘author-maintained’.

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

Contents

1	Micro-typography with pdf\TeX	4
2	Getting started	5
3	Options	6
3.1	Enabling the micro-typographic features	6
3.2	Character protrusion	7
3.3	Font expansion	8
3.4	Tracking/letterspacing	9
3.5	Miscellaneous options	9
3.6	Changing options later	10
4	Selecting fonts for micro-typography	10
5	Micro fine tuning	12
5.1	Character protrusion	13
5.2	Font expansion	14
5.3	Tracking	16
5.4	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	25
9	Hints and caveats	25
10	Contributions	28
11	Acknowledgments	28
12	References	29
13	Short history	29
14	Implementation	33
14.1	Preliminaries	34
	Debugging [35] Requirements [37] Declarations [40] Auxiliary macros [41] Compatibility [48]	
14.2	Font setup	53
	Protrusion [57] Expansion [64] Interword spacing (glue) [66] Additional kerning [68] Tracking [69] Disabling ligatures [79] Loading the configuration [79] Translating characters into slots [84] Hook into \LaTeX 's font selection [88] Context-sensitive setup [91]	

14.3	Configuration	94
	Font sets [94] Variants and aliases [99] Disabling ligatures [100]	
	Interaction with babel [101] Fine tuning [101] Character inheritance [107]	
	Permutation [109]	
14.4	Package options	112
	Declaring the options [112] Reading the configuration file [116] Hook	
	for other packages [117] Changing options later [118] Processing the	
	options [120]	
15	Configuration files	128
15.1	Font sets	128
15.2	Font variants and aliases	129
15.3	Interaction with babel	130
15.4	Note on admissible characters	130
15.5	Character inheritance	131
	OT1 [131] T1 [131] LY1 [132] OT4 [133] QX [133] T5 [134] Euro	
	symbols [135]	
15.6	Tracking	135
15.7	Font expansion	135
15.8	Character protrusion	137
	Normal [138] Italics [147] Small caps [156] Italic small caps [159] Text	
	companion [161] Computer Modern math [166] AMS symbols [169]	
	Euler [173] Euro symbols [177]	
15.9	Interword spacing	178
	Nonfrenchspacing [180]	
15.10	Additional kerning	181
	French [182] Turkish [182]	
16	Auxiliary file for micro fine tuning	183
A	Change history	185
B	Index	192
C	The L^AT_EX Project Public License	200

List of Tables

1	Availability of micro-typographic features	7
2	Predefined font sets	12
3	Fonts with tailored protrusion settings	21
4	Order for matching font attributes	82

1 Micro-typography with pdf \TeX

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

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

Protrusion	off
Expansion	off

Both features are enabled throughout this document.

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

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

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

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

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

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

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

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

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

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’. However, while space factors will influence all three parameters of interword space (or glue) by the same amount – the kerning, the maximum amount that the space may be stretched and the maximum amount that it may be shrunk – pdfTeX provides the possibility to modify these parameters independently from one another. Furthermore, the values may be set differently for each font. And, probably most importantly, the parameters may not only be increased but also decreased. This feature may enhance the appearance of paragraphs even more. Emphasis in the last sentence is on the word ‘may’: this extension is still highly experimental – in particular, only ending characters will currently have an influence on the interword space. Also, the settings that are shipped with microtype are but a first approximation, and I would welcome corrections and improvements very much. I suggest reading the reasoning behind the settings in section 15.9.

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

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

2 Getting started

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

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

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

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

3 Options

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

3.1 Enabling the micro-typographic features

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

expansion These are the main options to control the level of micro-typographic refinement which the fonts in your document should gain. By default, the package is moderately greedy: character protrusion will be enabled, font expansion will only be disabled in circumstances where pdf_TE_X cannot expand the fonts automatically, that is, if it is either too old (versions before 1.20) or if the output mode is DVI (see section 3.5). In other words, `microtype` will try to apply as much micro-typography as can safely be expected to work under the respective conditions (and it is usually not necessary to load the package with different options for PDF resp. DVI mode).

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

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

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

```
\usepackage{microtype}
```

T _E X engine			Micro-typographic features					
Engine	Version	Output	Protrusion	Expansion	(= auto)	Kerning	Spacing	Tracking
pdfT _E X	< 0.14f	DVI/PDF	⊘	⊘	⊘	⊘	⊘	⊘
	≥ 0.14f	DVI/PDF	★	☒	⊘	⊘	⊘	⊘
	≥ 1.20	DVI	★	☒	⊘	⊘	⊘	⊘
		PDF	★	★	★	⊘	⊘	⊘
	≥ 1.40	DVI	★	☒	⊘	☒	☒	⊘
		PDF	★	★	★	☒	☒	☒ ^a
luaT _E X	≥ 0.25	DVI	★	☒	⊘	⊘	⊘	⊘
		PDF	★	★	★	⊘	⊘	⊘

★ = enabled ☒ = not enabled ⊘ = not available ^a ≥ 1.40.4 recommended

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.

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

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

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

with typographical correctness – if you are using a large font that calls for more modest protrusion).

unit character, $\langle dimension \rangle$ character

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

3.3 Font expansion

auto true, false * true

As noted in chapter 1, the expanded versions of the fonts may be calculated automatically. This option is true by default provided that pdfTeX's version is found to be 1.20 or higher and the output mode is PDF; otherwise, it will be disabled. If auto is set to false, the fonts for all expansion steps must exist (with files called $\langle font\ name \rangle \pm \langle expansion\ value \rangle$, e. g., `cmr12+10`, as described in the [pdfTeX manual](#)).

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

stretch $\langle integer \rangle$ 20

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

step $\langle integer \rangle$ * 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), this option is by default set to 1, in order to allow pdfTeX to try 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 to increase the stretch and shrink limits (to, say, 30 instead of 20); however, the gain is limited since at the same time the average stretch variance will be decreased. Therefore, this option is by default set to false, so that all characters will be expanded by the same amount. See section 5.2 for a more detailed discussion.

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

³ 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.

3.4 Tracking/letterspacing

letterspace *<integer>* 100

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

3.5 Miscellaneous options

DVIoutput true, false * false

pdf \TeX is not only able to generate PDF output but can also spit out DVI files.⁴ The latter can be ordered with the option `DVIoutput`, which will set `\pdfoutput` to zero.

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

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

draft true, false false

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

verbose true, false, errors, silent false

Information on the settings used for each font will be written into the log file if you enable the `verbose` option. When `microtype` encounters a problem that is not fatal (e. g., an unknown character in the settings, or non-existent settings), it will by default only issue a warning and try to continue. Loading the package with `verbose=errors` will turn all warnings into errors, so that you can be sure that no problem will go unnoticed. If on the other hand you have investigated all warnings and decide to ignore them, you may silence `microtype` with `verbose=silent`.

babel true, false false

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

config *<file name>* microtype

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

⁴ Recent \TeX systems are using pdf \TeX as the default engine even for DVI output.

3.6 Changing options later

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

Inside the preamble, this command accepts all package options described above (except for `config`). In the document body, this command may be used to change the general settings of the micro-typographic extensions. It then accepts all options from section 3.1: `expansion`, `protrusion` and `activate`, which in turn may receive the values `true`, `false`, `compatibility` or `nocompatibility`, and `tracking`, `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 that are being used in the document, and a basic set of fonts will be subject to font expansion. You may want to customise which fonts should get the benefit of micro-typographic treatment. This can be achieved by declaring and activating ‘font sets’; these font sets are specified via font attributes that have to match.

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

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

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

```
\DeclareMicrotypeSet{basictext}
{ encoding = {OT1,T1,T2A,LY1,OT4,QX,T5},
  family   = {rm*,sf*},
  series    = {md*},
  size      = {normalsize,footnotesize,small,large}
}
```

If you now call

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

in the document’s preamble, only fonts in the text encodings OT1, T1, T2A, LY1, OT4, QX or T5, roman or sans serif families, normal (or ‘medium’) series, and in sizes called by `\normalsize`, `\footnotesize`, `\small` or `\large`, will be protruded. Math fonts, on the other hand, will not, since they are in another encoding. Neither will fonts in bold face, or huge fonts. Etc.

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

```
\DeclareMicrotypeSet{alltext}
{ encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
```

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

If a value is followed by an asterisk (like ‘rm*’ and ‘sf*’ in the first example), it does not designate an NFSS code, but will be translated into the document’s `\(value)default`, e. g., `\rmdefault`.⁵ 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 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 allowed for the font key. A single asterisk is equivalent to ‘*/*/*/*/*’, i. e., the normal font. Size selection commands are possible, too, however, ranges are not allowed.

Table 2 lists the nine predefined font sets. They may also be activated by passing their name to the feature options protrusion, expansion, tracking, 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.

⁵ 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	∅	∅	∅	∅	∅
alltext (allmath)	Text encodings, TS1 (OML, OMS, U)	∅	∅	∅	∅
basictext (basicmath)	Text encodings (OML, OMS)	$\rm*$, $\sf*$	$\md*$	∅	\normalsize , \footnotesize , \small , \large
smallcaps	Text encodings	∅	∅	$\sc*$	∅
footnotesize	Text encodings, TS1	∅	∅	∅	$-\small$
scriptsize	Text encodings, TS1	∅	∅	∅	$-\footnotesize$
normalfont	$\encoding*$	$\family*$	$\series*$	$\shape*$	\normalsize
'Text encodings' = OT1, T1, T2A, LY1, OT4, QX, T5					$\ldots*$ = $\ldots\text{default}$

`\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 used for character protrusion and additional kerning, the ‘basictext’ set for font expansion and interword spacing, and the ‘smallcaps’ set for tracking.

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

5 Micro fine tuning

Every character asks for a particular protrusion, 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.

The set of fonts to which the settings should apply is declared using the same syntax of *font axis* = *value list* pairs as for the command `\DeclareMicrotypeSet` (see section 4). The only difference is that asterisked values will be 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 both settings for the current family (say, T1/cmr//) and settings for italic fonts in the normal weight (T1//m/it/) exist, those for the cmr family would apply.⁶ The encoding must always match.

⁶ For the interested, table 4 on page 82 presents the exact order.

5.1 Character protrusion

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

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

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

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

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

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

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

Options:

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

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

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

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

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

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

unit By default, the protrusion factors are relative to the respective character's width. The `unit` option may be used to override this and make `microtype` regard all values in the list as thousandths of the specified width. Issuing, for instance, `'unit=1em'` would have the effect that a value of, say, 50 now results in the character being protruded by 5% of an em of the font (thus simulating the internal measuring of pdfTeX's `\lcode` and `\rcode` primitives). The default behaviour can be restored with `unit=character`.⁷

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

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

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

5.2 Font expansion

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

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

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

The *expansion settings* consist of *<character> = <expansion factor>* pairs. You may specify one number for each character, which determines the amount that a character may be expanded. The numbers denominate thousandths of the full expansion.

⁷ The `unit` option can even be passed globally to the package (cf. section 3.2). However, all provided settings are created under the assumption that the values are relative to the character width. Therefore, you should only change it if you are certain that the default settings will not be used in your document.

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

Options:

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

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

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

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

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

factor This option provides a different method to alter expansion settings for certain fonts, working around the restriction just mentioned. The factor option influences the expansion factors of all characters (in contrast to the overall stretchability) of the font. For instance, if you want the italic shape to be expanded less, you could declare:

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

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

5.3 Tracking

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

An important typographic technique – which was missing in T_EX for a long time – is the adjustment of tracking, i. e., the uniform addition or subtraction of letter space to/from all the characters in a font. For example, it is good typographic practice to slightly space out text set in all capitals or small capitals (as in this document). Legibility may also be improved by minimally increasing the tracking of smaller and decreasing that of larger type.⁸ The `\SetTracking` command allows to specify the tracking amount for different fonts or font sets. It will also be evaluated by the `\textls` command, which may be used for letterspacing shorter pieces of text (see section 7).

The *tracking amount* is specified in thousandths of 1 em (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 1 em.

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 1 em (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 0.166 em, 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 1 em (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

⁸ With full-featured fonts like Computer Modern, this is usually not necessary, though, since they come in optical sizes, and the tracking of the small-capitals font is already adjusted.

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

no ligatures As far as pdf \TeX is concerned, ligatures in letterspaced fonts would be constructed as usual, which may be advisable when changing the tracking by only a small amount. For larger letterspacing amounts, on the other hand, the normal letter space within ligatures would have displeasing effects. This key expects a comma-separated list of characters for which ligatures should be disabled; only the character that begins a ligature must be specified. If the key is given without a value, *all* ligatures of the font will be disabled. This is not recommended, however, since it also entails that kerning will be switched off.⁹ The default settings disable ligatures for the character ‘f’ only, i.e., ‘ff’, ‘fi’, ‘ffi’, etc.¹⁰ In exceptional situations, you can manually break up a ligature by inserting ‘{\kern0pt}’ resp. babel’s “| shortcut, or protect it by enclosing it in \lslig (see section 7).

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

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

and then write:

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

this is the (typographically dubious) outcome:

Stop stealing sheep!

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 0.45 em) immediately before the piece of text may *stretch* by 0.25 em and *shrink* by 0.15 em. 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 (0.08 em) is added. Furthermore, one *ligature* wasn’t broken up, because we neglected to specify the ‘s’ in the `no ligatures` key.

⁹ The inseparable connexion of ligatures and kerns is a limitation of \TeX that will not be lifted before the advent of lua \TeX .

¹⁰ With pdf \TeX versions older than 1.40.4, *all* ligatures, and hence all kerning, will be disabled. It is therefore recommended to use at least version 1.40.4.

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

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

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

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

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

5.4 Additional kerning

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

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

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

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

Options:

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

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

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

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

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

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

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

5.5 Interword spacing

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

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

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

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

Options:

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

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

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

the space inserted after a full stop would be doubled (technically speaking: $2 \times \text{\fontdimen 2}$), as would the maximum stretch and shrink amounts of the interword space (`\fontdimen 3` and `4`). 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, with the only exception that exactly one encoding must be specified. The inheritance lists are declared as pairs of *<base character>* = *<list of inheriting characters>*. Unless you are using a different encoding or a very peculiarly shaped font, there should be no need to change the default character inheritance settings.

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

5.7 Configuration files

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

If you embark on creating new settings for a font family, you should put them into a separate file, whose name must be: `mt-.cfg` (e. g., `mt-cmr.cfg`),

Table 3: Fonts with tailored protrusion settings

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

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

and may contain all commands described in the current section 5. These files will be loaded automatically if you are actually using the respective fonts. This package ships with configuration files for a number of font families. Table 3 lists them all.

`\DeclareMicrotypeVariants` {*{list of suffixes}*}

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

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

`\DeclareMicrotypeAlias` {*{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.¹¹ This command will load the file `'mt-.cfg'`.

6 Context-sensitive setup

The microtype package also allows to apply different micro-typographic settings to the fonts depending on the context 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**, **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}
```

¹¹ Font package authors might also want to have a look at the hook `\Microtype@Hook`, described in the implementation part, section 14.4.3.

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

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

`\textls*` While the tracking feature, described in section 5.3, will apply to sets of fonts, you may also want to letterspace shorter pieces of text, regardless of the font in which they are typeset.¹² For such ad-hoc letterspacing, microtype introduces two

¹² Letterspacing should be used cautiously; in particular, letterspacing lower-case text is held in abhorrence by honourable typographers. Unless you know what you are doing, you should probably only letterspace small-capitals or all-capitals. Another just cause may be emphasis in texts typeset in Fraktur fonts.

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 switch on letterspacing for all subsequent fonts until the end of the current group. The starred version of `\textls` does not add any extra kerning before or after the text, which may be useful, e. g., for section titles. By default, each character will be spaced out by $100/1000\text{ em} = 0.1\text{ em}$; this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option, with decreasing significance in this order.

`\lslig` `{\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 to solve this problem: either don’t disable the ‘s’ and/or ‘c’ ligatures and break those that need to be broken up by inserting ‘`\kern0pt`’ or babel’s “|” shortcut; or disable them and protect those ligatures that need to be protected by enclosing them in the `\lslig` command. So, the following two solutions have the same result (namely, “*Ausſichtsloſigkeit*”).

```
\SetTracking[no ligatures={f}]{encoding = LY, family = yfrak}{100}
\textfrak{\lsstyle Aus:s{\kern0pt}ichts:los{\kern0pt}igkeit}
```

```
\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{100}
\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 L^AT_EX, the `letterspace` package also works with `eplain` or even only `miniltx`: for use with `eplain`, load the package with `\usepackage` inside the `\beginpackages ... \endpackages` environment; with `miniltx` (which does not support package options) simply `\input letterspace.sty`.

8 Disabling ligatures

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

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

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

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

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

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

9 Hints and caveats

Use settings that match your font. Although the default settings should give reasonable results for most fonts, the particular font you happen to be using may have different character shapes that necessitate more or less protrusion or expansion. In particular, italic letter shapes may differ wildly in different fonts, hence I have decided against providing default protrusion settings for them. The file `test-microtype.tex` might be of some help when adjusting the protrusion settings for a font.

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

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

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

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

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

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

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

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. 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 L^AT_EX packages (except for pdfcp_{rot}). However, life isn't perfect, so problems are to be expected. Currently, I am aware of the following issues:

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

Possible error messages and how to get rid of them:

- ! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.
This error message will occur if you are trying to employ font expansion while creating DVI output. Remember, that *automatic* font expansion only works when running pdfT_EX in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your T_EX system.
- ! pdfT_EX error (font expansion): auto expansion is only possible with scalable fonts.
Automatic font expansion has been improved in pdfT_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_latex: font ptmr8r cannot be expanded (not an included Type1 font)
and the PDF viewer complains about a missing font, e. g., Adobe Reader thusly:
Could not find a font in the Resources dictionary - using Helvetica instead.
With pdfT_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 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 pdf_latexDownloadBase14 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 (pdfTeX versions older than 1.30).

When applying micro-typographic enhancement to a large document with a lot of fonts, pdfTeX may be running out of some kind of memory. It can be increased by setting the respective parameter to a larger value. For web2c-based systems, e. g., TeX Live, change the settings in texmf.cnf, for MiKTeX, in the file miktex.ini (2.4 or older) resp. pdflatex.ini (2.5 or newer).

- pdfTeX warning (font expansion): font should be expanded before its first use
This warning will occur with pdfTeX versions older than 1.40.4, if tracking *and* expansion is applied to a font. It is harmless and can be ignored.

10 Contributions

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

11 Acknowledgments

This package would be pointless if *Hàn Thế Thành* hadn’t created the pdfTeX programme in the first place, which introduced the micro-typographic extensions and made them available to the TeX world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004](#) and [Thành 2008](#) and elsewhere. I also thank him and the rest of the pdfTeX team 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 Bühmann* has suggested the possibility to specify ranges of font sizes, and resourcefully assisted in implementing this. He also came up with some good ideas for the management of complex configurations. *Ulrich Dirr* has made numerous suggestion, especially concerning the new extensions of interword spacing adjustment and additional character kerning. My thanks also go to *Maciej Eder* for contributing settings for the QX encoding, as well as to *Karl Karlsson* for providing settings for the Cyrillic T2A encoding.

I thank *Philipp Lehman* for adding to his csquotes package the possibility to restore the original meanings of all activated characters, thus allowing for these characters to be used in the configuration files. *Peter Wilson* kindly provided a hook

in his `ledmac`/`ledpar` packages, so that critical editions can finally also benefit from character protrusion.

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

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, *The pdf \TeX user manual*, 25 January 2007. (Available from CTAN at [/systems/pdftex/](#); latest version at <http://sarovar.org/projects/pdftex/>)

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

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

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

Melchior Franz, *The soul package*, 17 November 2003. (Available from CTAN at [/macros/latex/contrib/soul/](#)). See also Heiko Oberdiek’s extension of this package, `soulutf8`, which adds Unicode support. (Available from CTAN at [/macros/latex/contrib/oberdiek/](#))

13 Short history

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

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 \geq 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 only [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 \geq 1.40: tracking/letterspacing, adjustment of interword spacing (glue), and additional kerning (new commands \SetTracking, \SetExtraSpacing, \SetExtraKerning; new options ‘tracking’, ‘spacing’, ‘kerning’) [5.3, 5.5, 5.4]
- New commands \textls and \sstyle 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 of fonts (pdfTeX \geq 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
- When using the `ledmac` package, character protrusion will work for the first time ever (pdfTeX \geq 1.30)

1.7 (2005/03/23)

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

1.6 (2005/01/24)

- New option ‘factor’ to influence protrusion resp. expansion of all characters of a font or font set [3.2, 5]

- When pdf \TeX 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 \LaTeX 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`).

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

`lua`: Code for `luaTeX` (microtype only).

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

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

Used for – surprise! – debugging purposes.

`config`: Surrounds all configuration modules.

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

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

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

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

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

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

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

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

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

Contributed by *Harald Harders*.

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

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

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

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

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

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

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

`cfg-e`: Surrounds Euro symbol configurations.

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

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

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

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

And now for something completely different.

¹ `<package|letterspace>`

14.1 Preliminaries

`\MT@MT` This is us.

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

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

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

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

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

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

```

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

```

These commands also have a starred version.

```

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

```

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

```

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

```

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

```

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

```

`\MT@warning` Communicate.

```

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

```

14.1.1 Debugging

`\tracingmicrotype` Cases for `\tracingmicrotype`:

```

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

```

```

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

```

\tracingmicrotypeinpdf

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

Cases for \tracingmicrotypeinpdf:

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

```
97 \newcount\tracingmicrotypeinpdf
```

Let's see how it works ...

```
\tracingmicrotypeinpdf=2
```

```

\MT@pdf@annot
\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.

```

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

```
104 \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).

```

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

```

```

121 }
122 </debug>
123 </package>

```

14.1.2 Requirements

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

0: `miniltx`

1: `eplain`

2: `LATEX`

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

```

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

```

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

```

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

```

`\MT@pdftex@no` pdf_TE_X's features for which we provide an interface here haven't always been available, and some specifics have changed over time. Therefore, we have to test which pdf_TE_X we're using, if any. `\MT@pdftex@no` will be used throughout the package to respectively do the right thing.

Currently, we have to distinguish seven cases for pdf_TE_X:

0: not running pdf_TE_X

1: pdf_TE_X (< 0.14f)

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

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

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

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

6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`¹³; `\pdftracingfonts`; always e-TeX (≥ 1.40)

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

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

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

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

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

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

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

```
192 \def\MT@clear@options{%
193   <plain> \MT@requires@latex1{%
194     \AtEndOfPackage{\let\unprocessedoptions\relax}%
```

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

```

195 \let\CurrentOption\@empty
196 <plain> }\relax
197 }

198 \ifnum\MT@pdftex@no <
199 <package> 2
200 <letterspace> 6
201 \MT@warning@n1{You
202 \ifcase\MT@pdftex@no
203 don't seem to be using pdftex.\MessageBreak
204 ~\MT@MT' only works with pdftex.\MessageBreak
205 Try running ~pdflatex' instead of
206 ~\ifx\XeTeXversion\@undefined\else xe\fi latex'%
207 \else
208 are using a pdftex version older than
209 <package> 0.14f%
210 <letterspace> 1.40%
211 .\MessageBreak
212 ~\MT@MT' does not work with this version.\MessageBreak
213 Please install a newer version of pdftex%
214 \fi
215 }
216 \MT@clear@options\MT@restore@catcodes
217 \endinput\fi

```

Since luaTeX is included in TeX Live 2008, we now support it by default, even though it's still experimental. Letterspacing doesn't work at all yet, since luaTeX doesn't know the \letterspacefont command.

```

218 <lua>|letterspace>
219 \ifx\directlua\@undefined \else
220 \ifx\directlua\relax \else
221 <letterspace> \MT@error
222 <letterspace> \MT@warning@n1
223 {~\MT@MT'
224 <letterspace> only works with luatex if you generate%
225 <letterspace> doesn't currently work with luatex.%
226 \MessageBreak
227 <letterspace> the package with the ~lua' option%
228 <letterspace> Bye bye%
229 }
230 <letterspace> {}
231 <letterspace> \MT@clear@options\MT@restore@catcodes
232 <letterspace> \expandafter\expandafter\expandafter\endinput
233 \fi
234 \fi
235 </lua>|letterspace>

```

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

```

236 \RequirePackage{keyval}[1997/11/10]
237 <*package>

```

\MT@toks We need a token register.

```

238 \newtoks\MT@toks

```

\ifMT@if@ A scratch if.

```

239 \newif\ifMT@if@

```

14.1.3 Declarations

```

\ifMT@protrusion      These are the global switches ...
\ifMT@expansion 240 \newif\ifMT@protrusion
  \ifMT@auto 241 \newif\ifMT@expansion
    242 \newif\ifMT@auto
  \ifMT@selected 243 \newif\ifMT@selected
\ifMT@noligatures 244 \newif\ifMT@noligatures
  \ifMT@draft 245 \newif\ifMT@draft
  \ifMT@spacing 246 \newif\ifMT@spacing
  \ifMT@kerning 247 \newif\ifMT@kerning
  \ifMT@tracking 248 \newif\ifMT@tracking
  \ifMT@babel 249 \newif\ifMT@babel
  \MT@MT@babel ... and numbers.
  \MT@ex@level 250 \let\MT@pr@level\tw@
  \MT@pr@factor 251 \let\MT@ex@level\tw@
  252 \let\MT@pr@factor\@m
  \MT@ex@factor 253 \let\MT@ex@factor\@m
  \MT@sp@factor 254 \let\MT@sp@factor\@m
  \MT@kn@factor 255 \let\MT@kn@factor\@m

  \MT@pr@unit      Default unit for protrusion settings is character width, for spacing space, for kerning
  \MT@sp@unit      (and tracking) 1 em.
  \MT@kn@unit 256 \let\MT@pr@unit\@empty
  257 \let\MT@sp@unit\@m
  258 \def\MT@kn@unit{1em}

  \MT@stretch      Expansion settings.
  \MT@shrink 259 \let\MT@stretch\@m
  \MT@step 260 \let\MT@shrink \@m
  261 \let\MT@step \@m

  \MT@pr@min      Minimum and maximum values allowed by pdfTeX.
  \MT@pr@max 262 \def\MT@pr@min{-\@m}
  \MT@ex@min 263 \let\MT@pr@max\@m
  264 \let\MT@ex@min\z@
  \MT@ex@max 265 \let\MT@ex@max\@m
  \MT@sp@min 266 \def\MT@sp@min{-\@m}
  \MT@sp@max 267 \let\MT@sp@max\@m
  \MT@kn@min 268 \def\MT@kn@min{-\@m}
  \MT@kn@max 269 \let\MT@kn@max\@m
  \MT@tr@min 270 /package
  \MT@tr@max 271 \def\MT@tr@min{-\@m}
  272 \let\MT@tr@max\@m
  \MT@tr@max 273 *package

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

\MT@stretch@default      Default values for expansion.
\MT@shrink@default 275 \def\MT@stretch@default{20 }
\MT@step@default 276 \def\MT@shrink@default{20 }
  277 \def\MT@step@default{4 }

  \MT@letterspace      Default value for letterspacing (in thousandths of 1 em).
\MT@letterspace@default 278 /package
  279 \let\MT@letterspace\@m
  280 \def\MT@letterspace@default{100}
  281 *package

```


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

282 `\newif\ifMT@document`

14.1.4 Auxiliary macros

`\MT@maybe@etex` For definitions that depend on e-TeX features.

```
283 \ifcase 0%
284   \ifx\TeXversion\undefined 1\else
285   \ifx\TeXversion\relax 1\else
286   \ifcase\TeXversion 1\fi
287   \fi
288 \fi
289 \else
290   \catcode`\^^Q=9 \catcode`\^^X=14
291 \fi
292 <debug>\MT@info{n1}{0}{this is
293 <debug>^^Q not
294 <debug> etex}
```

`\MT@requires@pdftex` For definitions that depend on a particular pdfTeX version.

```
295 \def\MT@requires@pdftex#1{%
296   \ifnum\MT@pdftex@no<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
297 }
298 <debug>\MT@requires@pdftex6{\pdftracingfonts=1 }\relax
```

`\MT@requires@luatex` For definitions that depend on luaTeX.

```
299 <*lua>
300 \let\MT@requires@luatex\@secondoftwo
301 \ifx\directlua\undefined \else
302   \ifx\directlua\relax \else
303     \let\MT@requires@luatex\@firstoftwo
304   \fi
305 \fi
306 <debug>\MT@info{n10}{this is \MT@requires@luatex}{\not }luatex}
```

`\MT@lua` Communicate with lua. Beginning with luaTeX 0.36, `\directlua` no longer requires a state number. `\luatexversion` ought to have been enabled by the format.

```
307 \MT@requires@luatex{
308   \ifnum\luatexversion<36
309     \def\MT@lua{\directlua0}
310   \else
311     \def\MT@lua{\directlua}
312   \fi
313 }\relax
314 </lua>
315 </package>
```

`\MT@glet` The forgotten primitive.

316 `\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`

```
317 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
318 <*package>
319 \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 320 \def\MT@def@n{\MT@exp@cs\def}
321 \def\MT@gdef@n{\MT@exp@gcs\gdef}
```

```

\MT@edef@n    Its expanding versions.
\MT@xdef@n    322 </package>
               323 \def\MT@edef@n{\MT@exp@cs\edef}
               324 <*package>
               325 \def\MT@xdef@n{\MT@exp@gcs\xdef}

\MT@let@nc    \let a \csname sequence to a command.
\MT@glet@nc    326 \def\MT@let@nc{\MT@exp@cs\let}
               327 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}

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

\MT@let@nn    \let a \csname sequence to a \csname sequence.
\MT@glet@nn    329 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}
               330 \def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}

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

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

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

\MT@exp@two@n    Expand the next two tokens after <#1> once and enclose them in braces.
               336 \def\MT@exp@two@n#1#2#3{%
               337   \expandafter\expandafter\expandafter
               338   #1\expandafter\expandafter\expandafter
               339   {\expandafter#2\expandafter}\expandafter{#3}}

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

\MT@ifdefined@c@T    Wrapper for testing whether command resp. \csname sequence is defined. If we
\MT@ifdefined@c@TF    are running e-TeX, we will use its primitives \ifdefined and \ifcsname, which
\MT@ifdefined@n@T    decreases memory use substantially.
\MT@ifdefined@n@TF    340 \def\MT@ifdefined@c@T#1{%
                       341   ^^X \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
                       342   ^^Q \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
                       343 }
                       344 </package>
                       345 \def\MT@ifdefined@c@TF#1{%
                       346   ^^X \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
                       347   <package>^^Q \ifx#1\@undefined
                       348   <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
                       349 }
                       350 \def\MT@ifdefined@n@T#1{%
                       351   ^^X \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
                       352   <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
                       353   <package>^^Q \expandafter\@gobble\else\expandafter\@firstofone\fi
                       354 }
                       355 <*package>
                       356 \def\MT@ifdefined@n@TF#1{%
                       357   ^^X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
                       358   ^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
                       359   ^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
                       360 }

```

`\MT@detokenize@n` Translate a macro into a token list. With e-TeX, we can use `\detokenize`. We also need to remove the last trailing space; and only the last one – therefore the fiddling (and the `\string` isn't perfect, of course).

```
\MT@detokenize@c
\MT@rem@last@space
361 \def\MT@detokenize@n#1{%
362 ^^X \expandafter\MT@rem@last@space\detokenize{#1} \@nil
363 ^^Q \string#1%
364 }
365 \def\MT@detokenize@c#1{%
366 ^^X \MT@exp@one@n\MT@detokenize@n#1%
367 ^^Q \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil
368 }
369 \def\MT@rem@last@space#1 #2{#1%
370 \ifx\@nil#2\else \space
371 \expandafter\MT@rem@last@space\expandafter#2\fi
372 }
```

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

```
373 </package>
374 \begingroup
375 \catcode`\%=12
376 \catcode`\&=14
377 \gdef\MT@ifempty#1{%
378 \if %#1%&
379 \expandafter\@firstoftwo
380 \else
381 \expandafter\@secondoftwo
382 \fi
383 }
384 \endgroup
385 <*package>
```

In TeX Live 2009 all luaTeX primitives are prefixed with ‘`lua`’.

```
386 <*lua>
387 \MT@requires@luatex{
388 \MT@ifdefined@c@TF\luatexluaescapestring\relax
389 {\let\luatexluaescapestring\luaescapestring}
390 }\relax
391 </lua>
```

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

```
392 \MT@requires@pdftex6{
393 <*lua>
394 \MT@requires@luatex{
395 \def\MT@ifint#1{%
396 \csname \MT@lua{
397 if string.find("\luatexluaescapestring{#1}", "^-*[0-9]+ *$")
398 then tex.write("@firstoftwo")
399 else tex.write("@secondoftwo")
400 end}%
401 \endcsname
402 }
403 }{
404 </lua>
405 </package>
406 \def\MT@ifint#1{%
407 \ifcase\pdfmatch{^-*[0-9]+ *} {#1} \relax
408 \expandafter\@secondoftwo
409 \else
410 \expandafter\@firstoftwo
```

```

411     \fi
412   }
413   < *package >
414   < lua > }
415 }{
416   \def\MT@ifint#1{%
417     \if!\ifnum9<1#1!\else?\fi
418     \expandafter\@firstoftwo
419     \else
420     \expandafter\@secondoftwo
421     \fi
422   }
423 }

\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.)
424 \MT@requires@pdftex6{
425   < *lua >
426   \MT@requires@luatex{
427     \def\MT@ifdimen#1{%
428       \csname \MT@lua{
429         if (string.find("\luatexluaescapestring{#1}", "^-[0-9]+(\@percentchar a*) *$") or
430           string.find("\luatexluaescapestring{#1}", "^-[0-9]*[.][0-9]+(\@percentchar a*) *$"))
431           then tex.write("@firstoftwo")
432           else tex.write("@secondoftwo")
433         end}%
434       \endcsname
435     }
436   }{
437     < /lua >
438     \def\MT@ifdimen#1{%
439       \ifcase\pdfmatch{^[0-9]+([.][0-9]+)?|([.][0-9]+)%
440         (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
441       \expandafter\@secondoftwo
442       \else
443       \expandafter\@firstoftwo
444       \fi
445     }
446     < lua > }
447   }{
448     \def\MT@ifdimen#1{%
449       \setbox\z@=\hbox{%
450         \MT@count=1#1\relax
451         \ifnum\MT@count=\@ne
452           \aftergroup\@secondoftwo
453         \else
454           \aftergroup\@firstoftwo
455         \fi
456       }%
457     }
458   }

\MT@ifdim    Test floating point numbers.
459 \def\MT@ifdim#1#2#3{%
460   \ifdim #1\p@ #2 #3\p@
461     \expandafter\@firstoftwo
462   \else
463     \expandafter\@secondoftwo
464   \fi
465 }

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

```

```

466 \MT@requires@pdftex5{
467   {*lua}
468   \MT@requires@luatex{
469     \def\MT@ifstreq#1#2{%
470       \csname \MT@lua{
471         if "\luatexluaescapestring{#1}" == "\luatexluaescapestring{#2}"
472         then tex.write("@firstoftwo")
473         else tex.write("@secondoftwo")
474       }%
475     \endcsname
476   }
477 }{
478 {/lua}
479   \def\MT@ifstreq#1#2{%
480     \ifcase\pdfstrcmp{#1}{#2}\relax
481     \expandafter\@firstoftwo
482     \else
483     \expandafter\@secondoftwo
484     \fi
485   }
486 {lua} }
487 }{
488   \def\MT@ifstreq#1#2{%
489     \edef\MT@res@a{#1}%
490     \edef\MT@res@b{#2}%
491     \ifx\MT@res@a\MT@res@b
492     \expandafter\@firstoftwo
493     \else
494     \expandafter\@secondoftwo
495     \fi
496   }
497 }

```

\MT@xadd Add item to a list.

```

498 \def\MT@xadd#1#2{%
499   \ifx#1\relax
500     \xdef#1{#2}%
501   \else
502     \xdef#1{#1#2}%
503   \fi
504 }

```

\MT@xaddb Add item to the beginning.

```

505 \def\MT@xaddb#1#2{%
506   \ifx#1\relax
507     \xdef#1{#2}%
508   \else
509     \xdef#1{#2#1}%
510   \fi
511 }

```

\MT@map@clist@n Run <#2> on all elements of the comma list <#1>. This and the following is modelled after L^AT_EX3 commands.

```

\MT@map@clist@c
\MT@map@clist@
\MT@map@clist@ 512 {/package}
\MT@clist@function 513 \def\MT@map@clist@n#1#2{%
\MT@clist@break 514   \ifx\@empty#1\else
515     \def\MT@clist@function##1{#2}%
516     \MT@map@clist@#1,\@nil,\@nnil
517   \fi
518 }
519 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
520 \def\MT@map@clist@#1,%

```

```

521 \ifx\@nil#1%
522 \expandafter\MT@clist@break
523 \fi
524 \MT@clist@function{#1}%
525 \MT@map@clist@
526 }
527 \let\MT@clist@function\@gobble
528 \def\MT@clist@break#1\@nnil{}
529 (*package)

\MT@map@tlist@ 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@ 530 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break 531 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
532 \def\MT@map@tlist@#1#2{%
533 \ifx\@nnil#2\else
534 #1{#2}%
535 \expandafter\MT@map@tlist@
536 \expandafter#1%
537 \fi
538 }
539 \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 540 \newif\ifMT@inlist@
541 \def\MT@in@clist#1#2{%
542 \def\MT@res@a#1,#1,##2##3\@nnil{%
543 \ifx##2\empty
544 \MT@inlist@false
545 \else
546 \MT@inlist@true
547 \fi
548 }%
549 \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
550 }

\MT@rem@from@clist Remove item <#1> from comma list <#2>. This is basically \@removeelement from
ltnctrl.dtx. Using \pdfmatch and \pdflastmatch here would be really slow!
551 \def\MT@rem@from@clist#1#2{%
552 \def\MT@res@a#1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
553 \def\MT@res@b#1,\MT@res@b##2\MT@res@b{\ifx,##1\empty\else##1\fi}%
554 \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
555 }

\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.
556 \def\MT@in@tlist#1#2{%
557 \MT@inlist@false
558 \def\MT@res@a{#1}%
559 \MT@map@tlist@c#2\MT@in@tlist@
560 }
561 \def\MT@in@tlist@#1{%
562 \edef\MT@res@b{#1}%
563 \ifx\MT@res@a\MT@res@b
564 \MT@inlist@true
565 \expandafter\MT@tlist@break
566 \fi
567 }

\MT@in@rlist Test whether size \MT@size is in a list of ranges. Store the name of the list in
\MT@in@rlist@
\MT@in@rlist@@
\MT@size@name

```

```

\MT@size@name
568 \def\MT@in@rlist#1{%
569   \MT@inlist@false
570   \MT@map@tlist@c#1\MT@in@rlist@
571 }
572 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
573 \def\MT@in@rlist@@#1#2#3{%
574   \MT@ifdim{#2}=\m@ne{%
575     \MT@ifdim{#1}=\MT@size
576     \MT@inlist@true
577     \relax
578   }{%
579     \MT@ifdim\MT@size<{#1}\relax{%
580       \MT@ifdim\MT@size<{#2}%
581       \MT@inlist@true
582       \relax
583     }%
584   }%
585   \ifMT@inlist@
586     \def\MT@size@name{#3}%
587     \expandafter\MT@tlist@break
588   \fi
589 }

\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 590 </package>
591 \def\MT@loop#1\MT@repeat{%
592   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
593   \MT@iterate \let\MT@iterate\relax
594 }
595 \let\MT@repeat\fi

\MT@while@num Execute <#3> from <#1> up to (excluding) <#2> (much faster than LATEX's \@whilenum).
596 \def\MT@while@num#1#2#3{%
597   \@tempcnta#1\relax
598   \MT@loop #3%
599   \advance\@tempcnta \@ne
600   \ifnum\@tempcnta < #2\MT@repeat
601 }

\MT@do@font Execute <#1> 256 times.
602 \def\MT@do@font{\MT@while@num\z@\@ccclvi}
603 <package>

\MT@count Increment macro <#1> by one. Saves using up too many counters. The e-TEX way is
\MT@increment slightly faster.
604 \newcount\MT@count
605 \def\MT@increment#1{%
606   ^^X \edef#1{\number\numexpr #1 + 1\relax}%
607   ^^Q \MT@count=#1\relax
608   ^^Q \advance\MT@count \@ne
609   ^^Q \edef#1{\number\MT@count}%
610 }

\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.
611 \def\MT@scale#1#2#3{%

```

```

612 ^^Q \multiply #1 #2\relax
613 \ifnum #3 = \z@
614 ^^X #1=\numexpr #1 * #2\relax
615 \else
616 ^^X #1=\numexpr #1 * #2 / #3\relax
617 ^^Q \divide #1 #3\relax
618 \fi
619 }

```

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

```

\MT@abbr@pr@c 620 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 621 \def\MT@abbr@ex{expansion}
\MT@abbr@pr@inh 622 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@ex@inh 623 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh 624 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@n 625 \def\MT@abbr@ex@inh{expansion inheritance}
\MT@abbr@sp 626 \def\MT@abbr@n{noligatures}
\MT@abbr@sp@c 627 \def\MT@abbr@sp{spacing}
\MT@abbr@sp@inh 628 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@sp@inh 629 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@kn 630 \def\MT@abbr@kn{ Kerning}
\MT@abbr@kn@c 631 \def\MT@abbr@kn@c{ Kerning codes}
\MT@abbr@kn@c 632 \def\MT@abbr@kn@inh{ Kerning inheritance}
\MT@abbr@kn@inh 633 \def\MT@abbr@tr{tracking}
\MT@abbr@kn@inh 634 \def\MT@abbr@tr@c{tracking amount}
\MT@abbr@tr
\MT@rbb@protrusion These we also need the other way round.
\MT@rbb@tr@c
\MT@rbb@expansion 635 \def\MT@rbb@protrusion{pr}
\MT@rbb@spacing 636 \def\MT@rbb@expansion{ex}
\MT@rbb@kerning 637 \def\MT@rbb@spacing{sp}
\MT@rbb@tracking 638 \def\MT@rbb@kerning{kn}
\MT@rbb@tracking 639 \def\MT@rbb@tracking{tr}

```

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

```

\MT@features@long 640 \def\MT@features{pr,ex,sp,kn,tr}
641 \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 `\@tempa`, the type of list to ignore in `(#1)`, then comes the action.

```

642 \def\MT@is@feature#1{%
643 \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
644 \ifMT@inlist@
645 \expandafter\@firstofone
646 \else
647 \MT@error{\@tempa' is not an available micro-typographic\MessageBreak
648 feature. Ignoring #1}{Available features are: \MT@features@long'.}%
649 \expandafter\@gobble
650 \fi
651 }

```

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 14.4.5)

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

```
652 \ifl@aded{tex}{wordcount}{%
653   \MT@warning@nl{Detected the `wordcount' utility.\MessageBreak
654     Disabling `MT@MT', since it wouldn't work}%
655   \MT@clear@options\MT@restore@catcodes\endinput}\relax
```

`\MT@setup@` The setup is deferred until the end of the preamble. This has a couple of advantages: `\microtypesetup` can be used to change options later on in the preamble, and fonts don't have to be set up before microtype.

```
656 </package>
657 <plain>\MT@requires@latex1{
658 \let\MT@setup@ \@empty
```

`\MT@addto@setup` We use our private hook to have better control over the timing. This will also work with `eplain`, but not with `miniltx` alone.

```
659 \def\MT@addto@setup{\g@addto@macro\MT@setup@
```

Don't hesitate with `miniltx`.

```
660 <plain>){\let\MT@addto@setup\@firstofone}
```

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

```
661 \def\MT@with@package@T#1{\ifpackage@loaded{#1}\@firstofone\@gobble}
662 <*package>
```

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

```
663 \def\MT@with@babel@and@T#1{%
664   \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
665     \expandtwoargs\MT@in@clist{#1}
666     {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
667     \ifMT@inlist\expandafter\@gobble\fi
668   }@gobble
669 }
```

Don't load `letterspace`.

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

`\MT@ledmac@setup` The `ledmac` package first saves each paragraph in a box, from which it then splits 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 pdf_TE_X version 1.21b together with `ledpatch.sty` as of 2005/06/02 (v0.4), character protrusion will work at last.

Peter Wilson was so kind to provide the `\l@dunhbox@line` hook in `ledmac` to allow for protrusion. `\leftmarginkern` and `\rightmarginkern` are new primitives of pdf_TE_X 1.21b (aka. 1.30.0).

```
671 \MT@requires@pdftex5{
672   \def\MT@ledmac@setup{%
673     \ifMT@protrusion
674       \MT@ifdefined@c@TF\l@dunhbox@line{%
675         \MT@info@nl{Patching ledmac to enable character protrusion}%
676         \newdimen\MT@led@kern
677         \let\MT@led@unhbox@line\l@dunhbox@line
```

```

678 \renewcommand*{\l@dunhbox@line}[1]{%
679 \ifhbox##1%
680 \MT@led@kern=\rightmarginkern##1%
681 \kern\leftmarginkern##1%
682 \MT@led@unhbox@line##1%
683 \kern\MT@led@kern
684 \fi
685 }%
686 }{%
687 \MT@warning@nl{%
688 Character protrusion in paragraphs with line\MessageBreak
689 numbering will only work if you update ledmac}%
690 }%
691 \fi
692 }
693 }{
694 \def\MT@ledmac@setup{%
695 \ifMT@protrusion
696 \MT@warning@nl{%
697 The pdftex version you are using does not allow\MessageBreak
698 character protrusion in paragraphs with line\MessageBreak
699 numbering by the 'ledmac' package.\MessageBreak
700 Upgrade pdftex to version 1.30 or later}%
701 \fi
702 }
703 }

```

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

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

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

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

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

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

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

```

706 \MT@if@false
707 \MT@with@babel@and@T{spanish} \MT@if@true
708 \MT@with@babel@and@T{galician} \MT@if@true
709 \MT@with@babel@and@T{mexican} \MT@if@true
710 \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.)

```

711 \MT@with@package@T{csquotes}{%
712 \ifpackage@later{csquotes}{2005/05/11}\@disablequotes\relax}%

```

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

```

713 \MT@if@false
714 \MT@with@package@T{hyperref} \MT@if@true
715 \MT@with@package@T{tex4ht} \MT@if@true

```

```

716 \ifMT@if@MT@restore@p@h\fi
717 }

```

Check again at the end of the preamble.

```

718 </package>
719 \MT@addto@setup{%
720 <*package>

```

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

```

721 \MT@with@package@T{pdfcpot}{%
722 \MT@error{Detected the `pdfcpot' package!\MessageBreak
723 ~\MT@MT' and `pdfcpot' may not be used together}{%
724 The `pdfcpot' package provides an interface to character protrusion.\MessageBreak
725 So does the ~\MT@MT' package. Using both packages at the same\MessageBreak
726 time will almost certainly lead to undesired results. Have your choice!}%
727 }%
728 \MT@with@package@T{ledmac}\MT@ledmac@setup

```

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

```

729 \let\MT@setupfont@hook\empty
730 \MT@if@false
731 \MT@with@babel@and@T{spanish} \MT@if@true
732 \MT@with@babel@and@T{galician}\MT@if@true
733 \MT@with@babel@and@T{mexican} \MT@if@true
734 \ifMT@if@
735 \g@addto@macro\MT@setupfont@hook{%
736 \MT@ifdefined@c@T\percentsign{\let%\percentsign}}%
737 \fi
738 \MT@with@package@T{csquotes}{%
739 \ifpackage@later{csquotes}{2005/05/11}{%
740 \g@addto@macro\MT@setupfont@hook\@disablequotes
741 }{%
742 \MT@warning@n1{%
743 Should you receive warnings about unknown slot\MessageBreak
744 numbers, try upgrading the `csquotes' package}%
745 }%
746 }%

```

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.

```

747 \MT@if@false
748 </package>
749 <plain> \MT@requires@latex2{
750 \MT@with@package@T{hyperref}{%
751 \pdfstringdefDisableCommands{%
752 <*package>
753 \let\pickup@font\MT@orig@pickupfont
754 \let\textmicrotypecontext\@secondoftwo
755 \let\microtypecontext\@gobble
756 </package>
757 \def\lststyle{\pdfstringdefWarn\lststyle}%
758 \def\textls#1#{\pdfstringdefWarn\textls}%
759 }%
760 <package> \MT@if@true
761 }%
762 <plain> }\relax
763 <*package>
764 \MT@with@package@T{tex4ht}\MT@if@true
765 \ifMT@if@\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi

```

The listings package makes numbers and letters active,

```

766 \MT@with@package@T{listings}{%
767   \g@addto@macro\MT@cfg@catcodes{%
768     \MT@while@num{"30}{ "3A}{\catcode\@tempcnta 12\relax}%
769     \MT@while@num{"41}{ "5B}{\catcode\@tempcnta 11\relax}%
770     \MT@while@num{"61}{ "7B}{\catcode\@tempcnta 11\relax}%
771   }%

```

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

```

772   \g@addto@macro\MT@setupfont@hook{%
773     \catcode`\z@

```

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

```

774   \let\lst@ProcessLetter\empty
775 }%
776 }%

```

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.

```

777 </package>
778 <plain> \MT@requires@latex2{
779   \MT@with@package@T{soul}{%
780     \soulregister\lsstyle 0%
781     \soulregister\textls 1%
782   }%

```

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

```

783 <*plain>
784   {\ifx\SOUTL@\undefined\else
785     \soulregister\lsstyle 0%
786     \soulregister\textls 1%
787   \fi}%
788 </plain>
789 <*package>

```

Compatibility with the `pinyin` package (from CJK): disable `microtype` in `\py@macron`, which loads a different font for the accent. In older versions of `pinyin` (pre-4.6.0), `\py@macron` had only one argument.

```

790 \MT@with@package@T{pinyin}{%
791   \let\MT@orig@py@macron\py@macron
792   \ifpackageafter{pinyin}{2005/08/11}{% 4.6.0
793     \def\py@macron#1#2{%
794       \let\pickup@font\MT@orig@pickupfont
795       \MT@orig@py@macron{#1}{#2}%
796       \let\pickup@font\MT@pickupfont}%
797     }{%
798       \def\py@macron#1{%
799         \let\pickup@font\MT@orig@pickupfont
800         \MT@orig@py@macron{#1}%
801         \let\pickup@font\MT@pickupfont}%
802       }%
803     }%
804 </package>
805 }
806 <*package>

```

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

```
807 \expandafter\ifx\the\font\nullfont\normalfont\fi
```

14.2 Font setup

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

```
808 \def\MT@setupfont{\MT@setupfont@hook}
```

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

```
809 \MT@requires@pdftex7
```

```
810 {\g@addto@macro\MT@setupfont\MT@copy@font}\relax
```

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

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

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

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

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

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

```
815 \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.)

```
816 % \ifx\f@encoding\cf@encoding\else\@enc@update\fi
```

```
817 }
```

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

```
818 \MT@requires@pdftex6
```

```
819 {\g@addto@macro\MT@setupfont\MT@tracking}\relax
```

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

```
821 \MT@check@font
```

```
822 \ifMT@inlist@
```

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

```
824 \else
```

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

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

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

```
826 \MT@protrusion
```

```
827 \MT@expansion
```

```
828 }
```

Interword spacing and kerning (pdfTeX 1.40).

```
829 \MT@requires@pdftex6
```

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

Disable ligatures (pdfTeX 1.30).

```
831 \MT@requires@pdftex5
```

```
832 {\g@addto@macro\MT@setupfont\MT@noligatures}\relax
833 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
834 <debug>\MT@show@pdfannot1%
```

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

```
835 \MT@register@font
836 \fi
837 }
```

\MT@copy@font The new (1.40.4) \pdfcopyfont command allows to expand a font with different parameters, or to use more than one set of protrusion factors for a given font within one paragraph. It will be used when we find a context for \SetProtrusion or \SetExpansion in the preamble, or when the package has been loaded with the copyfonts option.

```
838 \let\MT@copy@font\relax
839 \MT@requires@pdftex7{
840 \def\MT@copy@font{%
```

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

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

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

```
842 \expandafter\ifx\MT@font@copy\relax
843 \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
844 \expandafter\ifx\MT@font@orig\relax
845 \MT@exp@two@c\MT@gl@et\MT@font@orig\font@name
846 \else
847 \MT@exp@two@c\let\font@name\MT@font@orig
848 \fi
849 \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
850 <debug>\MT@dinfol{creating new copy: \MT@font@copy}%
```

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

```
851 \MT@map@clist@c\MT@active@features{%
852 \MT@exp@cs\ifx\MT@font@name\MT@abbr@#1}\relax\else
853 \def\@tempa{##1}%
854 \MT@exp@cs\MT@map@tlist@c\MT@font@doc@contexts\MT@rem@from@list
855 \fi
856 }%
857 \fi
858 \MT@exp@two@c\let\MT@font\MT@font@copy
```

We only need the font identifier for letterspacing.

```
859 \let\font@name\MT@font@copy
```

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

```
860 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
861 }
```

\MT@rem@from@list

```
862 \def\MT@rem@from@list#1{%
863 \MT@exp@cs\ifx\MT@font@tempa @#1font@list}\relax\else
864 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
865 \MT@font \csname \MT@font@tempa @#1font@list\endcsname
866 \fi
867 }
868 }\relax
```

Here's the promised dirty trick for users of older pdfTeX versions, which works around the problem that the use of the same font with different expansion parameters is prohibited. If you do not want to create a clone of the font setup (this would require duplicating the tfm/vf files under a new name, and writing new fd files and map entries), you can load a minimally larger font for the paragraph in question. E. g., for a document typeset in 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@split@name Split up the font name ((#6) may be a protrusion/expansion context and/or a
\MT@encoding letterspacing amount).
\MT@family 869 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
\MT@series 870 \def\MT@encoding{#1}%
871 \def\MT@family {#2}%
\MT@shape 872 \def\MT@series {#3}%
873 \def\MT@shape {#4}%
\MT@size 874 \def\MT@size {#5}%
875 }

\MT@familyalias Alias family?
875 \MT@ifdefined@n@TF{MT@MT@family @alias}%
876 {\MT@let@cn\MT@familyalias{MT@MT@family @alias}}%
877 {\let\MT@familyalias\empty}%
878 }

\ifMT@do We check all features of the current font against the lists of the currently active
\MT@feat font set, and set \ifMT@do accordingly.
\MT@maybe@do 879 \newif\ifMT@do
880 \def\MT@maybe@do#1{%
  (but only if the feature isn't globally set to false)
881 \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).

882 \MT@dotrue
883 \edef\@tempa{\csname MT@#1\setname\endcsname}%
884 \MT@map@clist@n{font,encoding,family,series,shape,size}%
885 \MT@ifdefined@n@TF{MT@checklist@#1}%
886 {\csname MT@checklist@#1\endcsname}%
887 {\MT@checklist@{#1}}%
888 {#1}%
889 }%
```

```

890 \else
891 \MT@dofalse
892 \fi
893 \ifMT@do
    \MT@feat stores the current feature.
894 \def\MT@feat{#1}%
895 \csname MT@set@#1\codes\endcsname
896 \else
897 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
898 \fi
899 }

```

\MT@info@list

```

900 <debug>\def\MT@info@list#1#2#3{\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: #2
901 <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).

```

902 \def\MT@checklist@#1#2{%
903 <!debug> \MT@ifdefined@n@T
904 <debug> \MT@ifdefined@n@TF
905 {MT@#2list@#1@tempa}%

```

Begin a (masqueraded) \expandafter orgy to test whether the font attribute is in the list.

```

906 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
907 \csname MT@#1\expandafter\endcsname
908 \csname MT@#2list@#1@tempa\endcsname
909 \ifMT@inlist@
910 <debug>\MT@info@list{#2}{#1}{in}%
911 \MT@dotrue
912 \else
913 <debug>\MT@info@list{#2}{#1}{not in}%
914 \MT@dofalse
915 \expandafter\MT@clist@break
916 \fi
917 }%

```

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.

```

918 <debug> {\MT@info@list{#2}{#1}{}}%
919 }

```

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

```

920 \def\MT@checklist@family#1{%
921 <!debug> \MT@ifdefined@n@T
922 <debug> \MT@ifdefined@n@TF
923 {MT@#1list@family@tempa}%
924 \MT@exp@two@n\MT@in@clist
925 \MT@family{\csname MT@#1list@family@tempa\endcsname}%
926 \ifMT@inlist@
927 <debug>\MT@info@list{#1}{family}{in}%
928 \MT@dotrue
929 \else
930 <debug>\MT@info@list{#1}{family}{not in}%
931 \MT@dofalse
932 \ifx\MT@familyalias\empty \else
933 \MT@exp@two@n\MT@in@clist
934 \MT@familyalias{\csname MT@#1list@family@tempa\endcsname}%
935 \ifMT@inlist@
936 <debug>\MT@info@list{#1}{family alias}{in}%
937 \MT@dotrue

```



```

938 <debug>\else\MT@info@list{#1}{family alias}{not in}%
939     \fi
940     \fi
941     \fi
942     \ifMT@do \else
943         \expandafter\MT@clist@break
944     \fi
945 }%
946 <debug> {\MT@info@list{#1}{family}}}%
947 }

```

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

```

948 \def\MT@checklist@size#1{%
949 <!debug> \MT@ifdefined@n@T
950 <debug> \MT@ifdefined@n@TF
951     {MT@#1list@size@\@tempa}%
952     \MT@exp@cs\MT@in@rlist{MT@#1list@size@\@tempa}%
953     \ifMT@inlist@
954 <debug>\MT@info@list{#1}{size}{in}%
955     \MT@dotrue
956     \else
957 <debug>\MT@info@list{#1}{size}{not in}%
958     \MT@dofalse
959     \expandafter\MT@clist@break
960     \fi
961 }%
962 <debug> {\MT@info@list{#1}{size}}}%
963 }

```

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

```

964 \def\MT@checklist@font#1{%
965 <!debug> \MT@ifdefined@n@T
966 <debug> \MT@ifdefined@n@TF
967     {MT@#1list@font@\@tempa}%

```

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

```

968     \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
969     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
970     \@tempb \csname MT@#1list@font@\@tempa\endcsname
971     \ifMT@inlist@
972 <debug>\MT@info@list{#1}{font}{in}%
973     \expandafter\MT@clist@break
974     \else
975 <debug>\MT@info@list{#1}{font}{not in}%
976     \MT@dofalse
977     \fi
978 }%
979 <debug> {\MT@info@list{#1}{font}}}%
980 }

```

14.2.1 Protrusion

\MT@protrusion Set up for protrusion?

```

981 \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.

```

982 \def\MT@set@pr@codes{%

```

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

```
983 \MT@if@list@exists{%
984 \MT@get@font@dimen@six{%
985 \MT@get@opt
986 \MT@reset@pr@codes
```

Get the name of the inheritance list and parse it.

```
987 \MT@get@inh@list
```

Set an input encoding?

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

Load additional lists?

```
989 \MT@load@list\MT@pr@c@name
990 \MT@set@listname
```

Load the main list.

```
991 \MT@let@cn@tempc{MT@pr@c@\MT@pr@c@name}%
992 \expandafter\MT@set@codes\@tempc,\relax,%
993 }\MT@reset@pr@codes
994 }
```

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

```
995 \def\MT@get@font@dimen@six{%
996 \ifnum\fontdimen6\MT@font=\z@
997 \MT@warning@n1{%
998 Font '\MT@font' does not specify its\MessageBreak
999 \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
1000 \@nameuse{MT@abbr@\MT@feat} will not work with this font}%
1001 \expandafter\@gobble
1002 \else
1003 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
1004 \expandafter\@firstofone
1005 \fi
1006 }
```

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

```
1007 \def\MT@set@all@pr#1#2{%
1008 <debug>\MT@info@n1{3}{-- lp/rp: setting all to #1/#2}%
1009 \let\MT@temp\empty
1010 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lpcode\MT@font\@tempcnta=#1\relax}}%
1011 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rpcode\MT@font\@tempcnta=#2\relax}}%
1012 \MT@do@font\MT@temp
1013 }
```

\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.

```
1014 \def\MT@reset@pr@codes@\MT@set@all@pr\z@\z@
1015 \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.

```
1016 \def\MT@the@pr@code{\@tempcntb}
1017 \MT@requires@pdftex6{
1018 \def\MT@the@pr@code@tr{%
1019 \numexpr\@tempcntb+\MT@letterspace@/2\relax
1020 }
1021 }\relax
```

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

```
1022 \def\MT@set@codes#1,{%
1023   \ifx\relax#1\@empty\else
1024     \MT@split@codes #1==\relax
1025     \expandafter\MT@set@codes
1026   \fi
1027 }
```

`\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.

```
1028 \def\MT@split@codes#1=#2=#3\relax{%
1029   \def\@tempa{#1}%
1030   \ifx\@tempa\@empty \else
1031     \MT@get@slot
1032     \ifnum\MT@char > \m@ne
1033       \MT@get@char@unit
1034       \csname MT@\MT@feat @split@val\endcsname#2\relax
1035     \fi
1036   \fi
1037 }
```

`\MT@pr@split@val`

```
1038 \def\MT@pr@split@val#1,#2\relax{%
1039   \def\@tempb{#1}%
1040   \MT@ifempty\@tempb\relax{%
1041     \MT@scale@to@em
1042     \lcode\MT@font\MT@char=\MT@the@pr@code
1043     (debug)\MT@info{n}{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char: [#1]}%
1044   }%
1045   \def\@tempb{#2}%
1046   \MT@ifempty\@tempb\relax{%
1047     \MT@scale@to@em
1048     \rprcode\MT@font\MT@char=\MT@the@pr@code
1049     (debug)\MT@info{n}{4}{;;; rp (\MT@char): \number\rprcode\MT@font\MT@char: [#2]}%
1050   }%
```

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

```
1051 \MT@ifdefined@c@T\MT@pr@inh@name{%
1052   \MT@ifdefined@n@T{\MT@inh@\MT@pr@inh@name @\MT@char @}{%
1053     \MT@exp@cs\MT@map@tlist@c
1054     {\MT@inh@\MT@pr@inh@name @\MT@char @}%
1055     \MT@set@pr@heirs
1056   }%
1057 }%
```

`\MT@scale@to@em`

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

```

1059 \MT@requires@pdftex3{
1060   \def\MT@scale@to@em{%
1061     \@tempcntb=\MT@count\relax

```

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

```

1062   \MT@scale@\@tempcntb \@tempb \MT@dimen@six
1063   \ifnum\@tempcntb=\z@ \else
1064     \MT@scale@factor
1065   \fi
1066 }

```

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

```

1067 \def\MT@get@charwd{%
1068 ^X \MT@count=\fontcharwd\MT@font\MT@char\relax
1069 ^Q \setbox\z@=\hbox{\MT@font \char\MT@char}%
1070 ^Q \MT@count=\wd\z@
1071 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1072 }

```

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 `1 em = \fontdimen 6`.

```

1073 \MT@requires@pdftex3{
1074   \g@addto@macro\MT@get@charwd{%
1075     \MT@ifdefined@c@T\MT@letterspace@
1076     {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1077   }
1078 } \relax
1079 }

```

No adjustment with versions 0.14f and 0.14g.

```

1080 \def\MT@scale@to@em{%
1081   \MT@count=\@tempb\relax
1082   \ifnum\MT@count=\z@ \else
1083     \MT@scale@factor
1084   \fi
1085 }

```

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

```

1086 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1087 }

```

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

```

1088 \def\MT@get@font@dimen#1{%
1089   \ifnum\fontdimen#1\MT@font=\z@
1090     \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1091       \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1092       You should use a different 'unit' for \MT@curr@list@name}%
1093   \else
1094     \MT@count=\fontdimen#1\MT@font
1095   \fi
1096 }

```

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

```

1097 \def\MT@info@missing@char{%
1098   \MT@info@n1{Character '\the\MT@toks'
1099   ^X \iffontchar\MT@font\MT@char
1100     has a width of 0pt

```

```

1101 ^^X    \else is missing\fi
1102 ^^Q    \MessageBreak (it's probably missing)
1103    \MessageBreak in font `\'MT@font'.\MessageBreak
1104    Ignoring protrusion settings for this character}%
1105 }

```

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

```

1106 \def\MT@scale@factor{%
1107   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1108     \expandafter\MT@scale\expandafter \@tempcntb
1109     \csname MT@\MT@feat @factor@\endcsname \@m
1110   \fi
1111   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
1112     \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1113   \else
1114     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1115       \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1116     \fi
1117   \fi
1118 }

```

`\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.

```

1119 \def\MT@warn@code@too@large#1{%
1120   \@tempcnta=#1\relax
1121   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1122     \expandafter\MT@scale\expandafter\@tempcnta\expandafter
1123     \@m \csname MT@\MT@feat @factor@\endcsname
1124   \fi
1125   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1126   \MT@warning@nl{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
1127     is too large for character\MessageBreak
1128     `\'the\MT@toks' in \MT@curr@list@name.\MessageBreak
1129     Setting it to the maximum of \number\@tempcnta}%
1130   \@tempcntb=#1\relax
1131 }

```

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

```

1132 \def\MT@get@opt{%
1133   \MT@set@listname

```

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

```

\MT@sp@factor@ 1134 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 1135   \MT@let@nn{MT@\MT@feat @factor@}
1136   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1137   \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1138     \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1139   }{%
1140     \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1141   }%

```

`\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@ 1142 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1143   \MT@let@nn{MT@\MT@feat @unit@}%
1144   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1145   \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
1146     \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1147       relative to character widths}%

```

```

1148 \else
1149 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
1150 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1151 relative to width of space}%
1152 \fi
1153 \fi
1154 }{%
1155 \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
1156 }%

```

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

```

1157 \let\MT@get@char@unit\relax
1158 \let\MT@get@space@unit\@gobble
1159 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
1160 \let\MT@get@char@unit\MT@get@charwd
1161 \else
1162 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
1163 \let\MT@get@space@unit\MT@get@font@dimen
1164 \else
1165 \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit}%
1166 \fi
1167 \fi

```

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

```

1168 \MT@ifdefined@n@T{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
1169 \csname MT@preset@\MT@feat\endcsname
1170 \MT@let@nc{MT@reset@\MT@feat @codes}\relax
1171 }%
1172 }

```

\MT@get@unit If unit contains an em or ex, we use the corresponding \fontdimen to obtain the real size. Simply converting the em into points might give a wrong result, since the font probably isn't set up yet, so that these dimensions haven't been updated, either.

```

1173 \def\MT@get@unit#1{%
1174 \expandafter\MT@get@unit@#1 e!\@nil
1175 \ifx\x\@empty\else\let#1\x\fi
1176 \@defaultunits@tempdima#1 pt\relax\@nnil
1177 \ifdim\@tempdima=\z@
1178 \MT@warning@n1{%
1179 Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
1180 width. Setting factors of list \@nameuse{MT@\MT@feat @c@name}'\MessageBreak
1181 relative to character widths instead}%
1182 \let#1\@empty
1183 \let\MT@get@char@unit\MT@get@charwd
1184 \else
1185 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
1186 to \the\@tempdima}%
1187 \MT@count=\@tempdima\relax
1188 \fi
1189 }
1190 \def\MT@get@unit@#1e#2#3\@nil{%
1191 \ifx\#3\\\let\x\@empty \else
1192 \if m#2%
1193 \edef\x{#1\fontdimen6\MT@font}%
1194 \else
1195 \if x#2%
1196 \edef\x{#1\fontdimen5\MT@font}%
1197 \fi

```

```

1198     \fi
1199     \fi
1200 }

\MT@set@inputenc    The configurations may be under the regime of an input encoding.
1201 \def\MT@set@inputenc#1{%

\MT@cat    We remember the current category (c or inh), in case of warnings later.
1202     \def\MT@cat{#1}%

1203     \edef\@tempa{\MT@MT@feat @#1\csname MT@MT@feat @#1\name\endcsname @inputenc}%
1204     \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1205 }

\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.
1206 \MT@addto@setup{%
1207     \ifpackageloaded{inputenc}{%
1208         \ifpackageafter{inputenc}{2006/02/22}{%
1209             \def\MT@set@inputenc{%
1210                 \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1211                 \MT@load@inputenc
1212             }%
1213         }%
1214         \let\MT@set@inputenc\MT@load@inputenc
1215     }%
1216 }{%
1217     \def\MT@set@inputenc{%
1218         \MT@warning@n{Key 'inputenc' used in \MT@curr@list@name, but the 'inputenc'
1219             \MessageBreak package isn't loaded. Ignoring input encoding}%
1220     }%
1221 }%
1222 }

\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.
1223 \def\MT@load@inputenc{%
1224     \MT@cfg@catcodes
1225     <debug>\MT@info@n{1}{loading input encoding: \@nameuse{\@tempa}}%
1226     \inputencoding{\@nameuse{\@tempa}}%
1227 }

\MT@set@pr@heirs    Set the inheriting characters.
1228 \def\MT@set@pr@heirs#1{%
1229     \lcode\MT@font#1=\lcode\MT@font\MT@char
1230     \rcode\MT@font#1=\rcode\MT@font\MT@char
1231     <debug>\MT@info@n{2}{-- heir of \MT@char: #1}%
1232     <debug>\MT@info@n{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
1233     <debug>                                     \number\rcode\MT@font\MT@char}%
1234 }

\MT@preset@pr    Preset characters. Presetting them relative to their widths is not allowed.
\MT@preset@pr@ 1235 \def\MT@preset@pr{%
1236     \expandafter\expandafter\expandafter\MT@preset@pr@
1237     \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1238 }
1239 \def\MT@preset@pr@#1,#2\@nil{%
1240     \ifx\MT@pr@unit@\@empty
1241         \MT@warn@preset@twidth{pr}%
1242         \let\MT@preset@aux\MT@preset@aux@factor
1243     \else
1244         \def\MT@preset@aux{\MT@preset@aux@space2}%

```

```

1245 \fi
1246 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1247 \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1248 \MT@set@all@pr\@tempa\@tempb
1249 }

```

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

```

\MT@preset@aux@factor 1250 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 1251 \@tempcntb=#1\relax
1252 \MT@scale@factor
1253 \edef#2{\number\@tempcntb}%
1254 }
1255 \def\MT@preset@aux@space#1#2#3{%
1256 \def\@tempb{#2}%
1257 \MT@get@space@unit#1%
1258 \MT@scale@to@em
1259 \edef#3{\number\@tempcntb}%
1260 }

\MT@warn@preset@tewidth
1261 \def\MT@warn@preset@tewidth#1{%
1262 \MT@warning@n1{%
1263 Cannot preset characters relative to their widths\MessageBreak
1264 for \@nameuse{MT@abbr#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1265 \MessageBreak relative to lem instead}%
1266 }

```

14.2.2 Expansion

\MT@expansion Set up for expansion?

```
1267 \def\MT@expansion{\MT@maybe@do{ex}}
```

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

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

```

1268 \def\MT@set@ex@codes@s{%
1269 \MT@if@list@exists{%
1270 \MT@get@ex@opt
1271 \let\MT@get@char@unit\relax
1272 \MT@reset@ef@codes
1273 \MT@get@inh@list
1274 \MT@set@inputenc{c}%
1275 \MT@load@list\MT@ex@cc@name
1276 \MT@set@listname
1277 \MT@let@cn\@tempc{MT@ex@c@\MT@ex@c@name}%
1278 \expandafter\MT@set@codes\@tempc,\relax,%
1279 \MT@expandfont
1280 }\relax
1281 }

```

\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.

```

1282 \newif\ifMT@nonselected
1283 \def\MT@set@ex@codes@n{%
1284 \MT@nonselectedtrue

```



```

1285 \MT@if@list@exists
1286 \MT@get@ex@opt
1287 {%
1288 \let\MT@stretch@ \MT@stretch
1289 \let\MT@shrink@ \MT@shrink
1290 \let\MT@step@ \MT@step
1291 \let\MT@auto@ \MT@auto
1292 \let\MT@ex@factor@ \MT@ex@factor
1293}%
1294 \MT@reset@ef@codes
1295 \MT@expandfont
1296 \MT@nonselectedfalse
1297 }

```

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

```
1298 \let\MT@set@ex@codes\MT@set@ex@codes@n
```

`\MT@expandfont` Expand the font.

```

1299 \def\MT@expandfont{%
1300 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1301 }

```

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

```

1302 \def\MT@set@all@ex#1{%
1303 <debug>\MT@info{n1}{3}{-- ex: setting all to \number#1}%
1304 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1305 }
1306 \def\MT@reset@ef@codes@{\MT@set@all@ex\MT@ex@factor@}

```

`\MT@reset@ef@codes` However, this is only necessary for versions prior to 1.20.

```

1307 \MT@requires@pdftex4{
1308 \def\MT@reset@ef@codes{%
1309 \ifnum\MT@ex@factor@=\@m \else
1310 \MT@reset@ef@codes@
1311 \fi
1312 }
1313 }{
1314 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1315 }

```

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

```

1316 \def\MT@ex@split@val#1\relax{%
1317 \@tempcntb=#1\relax

```

Take an optional factor into account.

```

1318 \ifnum\MT@ex@factor@=\@m \else
1319 \MT@scale\@tempcntb \MT@ex@factor@ \@m
1320 \fi
1321 \ifnum\@tempcntb > \MT@ex@max
1322 \MT@warn@ex@too@large\MT@ex@max
1323 \else
1324 \ifnum\@tempcntb < \MT@ex@min
1325 \MT@warn@ex@too@large\MT@ex@min
1326 \fi
1327 \fi
1328 \efcode\MT@font\MT@char=\@tempcntb
1329 <debug>\MT@info{n1}{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%

```

Heirs, heirs, I love thy heirs.

```

1330 \MT@if@defined@c@T\MT@ex@inh@name{%
1331 \MT@if@defined@n@T\MT@inh@MT@ex@inh@name @\MT@char @}{%

```

```

1332      \MT@exp@cs\MT@map@tlist@c{MT@inh@MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
1333    }%
1334  }%
1335 }

\MT@warn@ex@too@large
1336 \def\MT@warn@ex@too@large#1{%
1337   \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1338   character\MessageBreak ``the\MT@toks' in \MT@curr@list@name.\MessageBreak
1339   Setting it to the maximum of \number#1}%
1340   \@tempcntb=#1\relax
1341 }

\MT@get@ex@opt      Apply different values to this font?
\MT@ex@factor@ 1342 \def\MT@get@ex@opt{%
\MT@stretch@ 1343   \MT@set@listname
1344   \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1345   \MT@let@cn\MT@ex@factor@{MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1346   \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor/1000}%
\MT@auto@ 1347   }%
1348   \let\MT@ex@factor@\MT@ex@factor
1349   }%
1350   \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1351   \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1352   \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
1353   \def\@tempa{autoexpand}%
1354   \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1355   \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @preset}{%
1356     \MT@preset@ex
1357     \let\MT@reset@ef@codes\relax
1358   }%
1359 }

\MT@get@ex@opt@
1360 \def\MT@get@ex@opt@#1#2{%
1361   \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}{%
1362     \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
1363     \MT@vinfo{... : #2}%
1364   }%
1365   \MT@let@nn{MT@#1@}{MT@#1}%
1366   }%
1367 }

\MT@set@ex@heirs
1368 \def\MT@set@ex@heirs#1{%
1369   \efcode\MT@font#1=\efcode\MT@font\MT@char
1370   <debug>\MT@dinfoln{2}{-- heir of \MT@char: #1}%
1371   <debug>\MT@dinfoln{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1372 }

\MT@preset@ex
1373 \def\MT@preset@ex{%
1374   \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
1375   \MT@scale@factor
1376   \MT@set@all@ex@tempcntb
1377 }

```

14.2.3 Interword spacing (glue)

\MT@spacing Adjustment of interword spacing?

```
1378 \MT@requires@pdftex6{
```

```

1379 \def\MT@spacing{\MT@maybe@do{sp}}

\MT@set@sp@codes      This is all the same.
1380 \def\MT@set@sp@codes{%
1381   \MT@if@list@exists{%
1382     \MT@get@font@dimen@six{%
1383       \MT@get@opt
1384       \MT@reset@sp@codes
1385       \MT@get@inh@list
1386       \MT@set@inputenc{c}%
1387       \MT@load@list\MT@sp@cc@name
1388       \MT@set@listname
1389       \MT@let@cn\@tempc{\MT@sp@cc\MT@sp@cc@name}%
1390       \expandafter\MT@set@codes\@tempc,\relax,}%
1391     }\MT@reset@sp@codes
1392   }

\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).
1393 \def\MT@sp@split@val#1,#2,#3\relax{%
1394   \def\@tempb{#1}%
1395   \MT@ifempty\@tempb\relax{%
1396     \MT@get@space@unit2%
1397     \MT@scale@to@em
1398     \knbscode\MT@font\MT@char=\@tempcntb
1399   (debug)\MT@info@n1{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1400   }%
1401   \def\@tempb{#2}%
1402   \MT@ifempty\@tempb\relax{%
1403     \MT@get@space@unit3%
1404     \MT@scale@to@em
1405     \stbscode\MT@font\MT@char=\@tempcntb
1406   (debug)\MT@info@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1407   }%
1408   \def\@tempb{#3}%
1409   \MT@ifempty\@tempb\relax{%
1410     \MT@get@space@unit4%
1411     \MT@scale@to@em
1412     \shbscode\MT@font\MT@char=\@tempcntb
1413   (debug)\MT@info@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1414   }%
1415   \MT@ifdefined@c@T\MT@sp@inh@name{%
1416     \MT@ifdefined@nT{\MT@inh@\MT@sp@inh@name @\MT@char @}{%
1417       \MT@exp@cs\MT@map@tlist@c{\MT@inh@\MT@sp@inh@name @\MT@char @}\MT@set@sp@heirs
1418     }%
1419   }%
1420 }

\MT@set@sp@heirs
1421 \def\MT@set@sp@heirs#1{%
1422   \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1423   \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1424   \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1425   (debug)\MT@info@n1{2}{-- heir of \MT@char: #1}%
1426   (debug)\MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1427   (debug)          \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1428 }

\MT@set@all@sp
\MT@reset@sp@codes 1429 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1430 (debug)\MT@info@n1{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1431   \let\MT@temp\empty

```

```

1432 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\kbscode\MT@font\@tempcnta=#1\relax}}%
1433 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1434 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1435 \MT@do@font\MT@temp
1436 }
1437 \def\MT@reset@sp@codes@{\MT@set@all@sp@z@z@z@}
1438 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1439 \def\MT@preset@sp{%
1440 \expandafter\expandafter\expandafter\MT@preset@sp@
1441 \csname MT@sp@ec\MT@sp@ec@name @preset\endcsname\@nil
1442 }
1443 \def\MT@preset@sp@#1,#2,#3\@nil{%
1444 \ifx\MT@sp@unit@\empty
1445 \MT@warn@preset@t@width{sp}%
1446 \MT@ifempty{#1}{\let\@tempa\empty}{\MT@preset@aux@factor{#1}\@tempa}%
1447 \MT@ifempty{#2}{\let\@tempc\empty}{\MT@preset@aux@factor{#2}\@tempc}%
1448 \MT@ifempty{#3}{\let\@tempb\empty}{\MT@preset@aux@factor{#3}\@tempb}%
1449 \else
1450 \MT@ifempty{#1}{\let\@tempa\empty}{\MT@preset@aux@space2{#1}\@tempa}%
1451 \MT@ifempty{#2}{\let\@tempc\empty}{\MT@preset@aux@space3{#2}\@tempc}%
1452 \MT@ifempty{#3}{\let\@tempb\empty}{\MT@preset@aux@space4{#3}\@tempb}%
1453 \fi
1454 \MT@set@all@sp\@tempa\@tempc\@tempb
1455 }
1456 }\relax

```

14.2.4 Additional kerning

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

```

1457 \MT@requires@pdftex6{
1458 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

1459 \def\MT@set@kn@codes{%
1460 \MT@if@list@exists{%
1461 \MT@get@font@dimen@six{%
1462 \MT@get@opt
1463 \MT@reset@kn@codes
1464 \MT@get@inh@list
1465 \MT@set@inputenc{c}%
1466 \MT@load@list\MT@kn@c@name
1467 \MT@set@listname
1468 \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
1469 \expandafter\MT@set@codes\@tempc,\relax,%
1470 }\MT@reset@kn@codes
1471 }

```

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

```

1472 \def\MT@kn@split@val#1,#2\relax{%
1473 \def\@tempb{#1}%
1474 \MT@ifempty\@tempb\relax{%
1475 \MT@get@space@unit2%
1476 \MT@scale@to@em
1477 \knbscode\MT@font\MT@char=\@tempcntb
1478 (debug)\MT@info{n1}{4}{;;; knbc (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1479 }%
1480 \def\@tempb{#2}%
1481 \MT@ifempty\@tempb\relax{%
1482 \MT@get@space@unit2%

```

```

1483 \MT@scale@to@em
1484 \knaccode\MT@font\MT@char=\@tempcntb
1485 (debug)\MT@info@n1{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1486 }%
1487 \MT@ifdefined@c@T\MT@kn@inh@name{%
1488 \MT@ifdefined@n@T\MT@inh@MT@kn@inh@name @\MT@char @}%
1489 \MT@exp@cs\MT@map@tlist@c\MT@inh@MT@kn@inh@name @\MT@char @\MT@set@kn@heirs
1490 }%
1491 }%
1492 }

\MT@set@kn@heirs
1493 \def\MT@set@kn@heirs#1{%
1494 \knbccode\MT@font#1=\knbccode\MT@font\MT@char
1495 \knaccode\MT@font#1=\knaccode\MT@font\MT@char
1496 (debug)\MT@info@n1{2}{-- heir of \MT@char: #1}%
1497 (debug)\MT@info@n1{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char/%
1498 (debug) \number\knaccode\MT@font\MT@char}%
1499 }

\MT@set@all@kn
\MT@reset@kn@codes 1500 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1501 (debug)\MT@info@n1{3}{-- knac/knbc: setting all to #1/#2}%
1502 \let\MT@temp\@empty
1503 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbccode\MT@font\@tempcnta=#1\relax}}%
1504 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knaccode\MT@font\@tempcnta=#2\relax}}%
1505 \MT@do@font\MT@temp
1506 }
1507 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1508 \let\MT@reset@kn@codes\relax

\MT@preset@kn
\MT@preset@kn@ 1509 \def\MT@preset@kn{%
1510 \expandafter\expandafter\expandafter\MT@preset@kn@
1511 \c@name MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1512 }
1513 \def\MT@preset@kn@#1,#2\@nil{%
1514 \ifx\MT@kn@unit@\@empty
1515 \MT@warn@preset@t@width{kn}%
1516 \let\MT@preset@aux\MT@preset@aux@factor
1517 \else
1518 \def\MT@preset@aux{\MT@preset@aux@space2}%
1519 \fi
1520 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1521 \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1522 \MT@set@all@kn\@tempa\@tempb
1523 }
1524 }\relax

```

14.2.5 Tracking

This only works with pdf_T_EX 1.40.

```

1525 \MT@requires@pdftex6{

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

\MT@tr@font@list 1526 \let\MT@tr@font@list\@empty
1527 \def\MT@tracking@{%
1528 \MT@exp@one@n\MT@in@c@list\MT@font\MT@tr@font@list
1529 \ifMT@in@list@else
1530 \MT@maybe@do{tr}%

```

```

1531 \ifMT@do\else
1532 \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1533 \fi
1534 \fi
1535 }
1536 </package>
1537 \let\MT@tracking
1538 <package> \MT@tracking@
1539 <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.

```

1540 \def\MT@set@tr@codes{%
1541 <*package>
1542 \MT@vinfo{Tracking font `\' \MT@font'\on@line}%
1543 \MT@get@font@dimen@six{%
1544 \MT@if@list@exists
1545 \MT@get@tr@opt
1546 \relax
1547 </package>
1548 \MT@if@defined@c@TF\MT@letterspace@ \relax{\let\MT@letterspace@\MT@letterspace}%
1549 \ifnum\MT@letterspace@=\z@

```

Zero tracking requires special treatment.

```

1550 \MT@set@tr@zero
1551 \else
1552 <package> \MT@vinfo{... Tracking by \number\MT@letterspace@}%

```

Letterspacing only works in PDF mode.

```

1553 \MT@warn@tracking@DVI

```

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

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

```

1554 \xdef\MT@lsfont{\csname\expandafter\string\font@name
1555 \number\MT@letterspace@ ls\endcsname}%
1556 \expandafter\ifx\MT@lsfont\relax
1557 <debug>\MT@info@nl{1}{... new letterspacing instance}%

```

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

```

1558 \MT@get@ls@basefont
1559 \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@

```

Scale interword spacing (not configurable in letterspace).

```

1560 <*package>
1561 \MT@if@defined@c@TF\MT@tr@ispace
1562 {\let\@tempa\MT@tr@ispace}%
1563 {\edef\@tempa{\MT@letterspace@*,,}}%
1564 \MT@if@defined@c@TF\MT@tr@ospace
1565 {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1566 {\edef\@tempa{\@tempa,,,}}%
1567 \expandafter\MT@tr@set@space\@tempa,%
1568 </package>
1569 <*letterspace>
1570 % spacing = {<letterspace amount>*,,}
1571 \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1572 * \fontdimen2\MT@lsfont/1000\relax
1573 </letterspace>

```

Adjust outer kerning (microtype only).

```
1574 (*package)
1575 \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
1576 \expandafter\MT@tr@set@okern\@tempa,%
```

Disable ligatures (not configurable in letterspace).

```
1577 \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1578 (/package)
1579 (*letterspace)
1580 % no ligatures = {f}
1581 \tagcode\MT@lsfont`f=\m@ne
1582 (/letterspace)
```

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

```
1583 (debug)\MT@info@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%
1584 \MT@do@font{\lcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
1585 \rprcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
1586 (package) \let\MT@the@pr@code\MT@the@pr@code@tr
1587 \fi
```

Finally, let the letterspaced font propagate.

```
1588 \aftergroup\MT@set@lsfont
1589 (package) \let\MT@font\MT@lsfont
```

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

```
\MT@curr@ls 1590 \xdef\MT@set@curr@ls{\def\@noexpand\MT@curr@ls{\MT@letterspace@}}%
1591 \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.

```
1592 (*package)
1593 \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
1594 \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1595 \MT@tr@outer@l
1596 (/package)
```

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

```
1597 \ifx\MT@ls@adjust\@empty
1598 (letterspace) % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}
1599 \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1600 \MT@ls@outer@k
1601 (*letterspace)
1602 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1603 \aftergroup\aftergroup\aftergroup\MT@ls@aftergroup
1604 (/letterspace)
```

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

```
1605 (*package)
1606 \else
1607 \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
1608 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1609 \ifdim\MT@outer@kern=z@\else \MT@ls@outer@k \fi
1610 \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
1611 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1612 (/package)
1613 \fi
1614 (*package)
```

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

```

1615 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1616 \aftergroup\aftergroup\aftergroup\MT@ls@aftergroup
1617 </package>
1618 \fi
1619 <package> }%
1620 }

```

`\MT@ls@aftergroup` Stuff to be done after the letterspace group. The `letterspace` package only adjusts the kerning.

```

1621 <letterspace>\def\MT@ls@aftergroup{\MT@set@curr@ok\MT@ls@outer@k}

```

microtype also adjusts spacing. If `\tikz@expandcount` is greater than zero, we're inside or at the end of a `tikz` node, where we don't want to do anything, lest we disturb `tikz`.

```

1622 <*package>
1623 \MT@addto@setup{%
1624 \ifpackageloaded{tikz}
1625 {\def\MT@ls@aftergroup{%
1626 \ifnum\tikz@expandcount>\z@ \else
1627 \MT@set@curr@os\MT@set@curr@ok\expandafter\MT@tr@outer@r\fi}}
1628 {\def\MT@ls@aftergroup{\MT@set@curr@os\MT@set@curr@ok\MT@tr@outer@r}}

```

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

```

1629 \def\MT@get@tr@opt{%
1630 \MT@set@listname
1631 \MT@ifdefined@n@T{\MT@tr@cc@\MT@tr@cc@name}{%
1632 \MT@let@cn\MT@letterspace{\MT@tr@cc@\MT@tr@cc@name}%

```

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

```

1633 \MT@ifdefined@n@T{\MT@tr@cc@\MT@tr@cc@name @unit}{%
1634 \MT@let@cn\MT@tr@unit@\MT@tr@cc@\MT@tr@cc@name @unit}%
1635 \ifdim\MT@tr@unit@=1em
1636 \let\MT@tr@unit@\undefined
1637 \else
1638 \MT@let@cn@tempb{\MT@tr@cc@\MT@tr@cc@name}%
1639 \MT@get@unit\MT@tr@unit@
1640 \let\MT@tr@factor@\@m
1641 \MT@scale@to@em
1642 \edef\MT@letterspace{\number\@tempcntb}%
1643 \fi
1644 }%
1645 }%

```

`\MT@tr@ispace` Adjust interword spacing.

```

\MT@tr@ospace 1646 \MT@get@tr@opt@{spacing} {ispace}%
1647 \MT@get@tr@opt@{outerspacing}{ospace}%

```

`\MT@tr@okern` Adjust outer kerning.

```

1648 \MT@get@tr@opt@{outerkerning}{okern}%

```

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

```

1649 \MT@get@tr@opt@{noligatures} {ligatures}%
1650 }

```

`\MT@get@tr@opt@`

```

1651 \def\MT@get@tr@opt@#1#2{%
1652 \MT@ifdefined@n@T{\MT@tr@cc@\MT@tr@cc@name @#1}%
1653 {\MT@let@nn{\MT@tr@cc@#2}{\MT@tr@cc@\MT@tr@cc@name @#1}}%

```



```

1654 }
1655 </package>

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

1656 <plain>\MT@requires@latex2{
1657 \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).

1658 \DeclareRobustCommand\lsstyle{%
1659   \not@math@alphabet\lsstyle\textls
1660 <package> \def\MT@feat{tr}%
1661 \let\MT@tracking\MT@set@tr@codes
1662 \selectfont
1663 }

```

Now the definitions for the letterspace package with plain T_EX.

```

1664 <*plain>
1665 }{
1666 \def\MT@set@lsfont{\MT@lsfont}
1667 \def\lsstyle{%
1668   \begingroup
1669   \escapechar\m@ne
1670   \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1671   \MT@set@tr@codes
1672   \endgroup
1673 }
1674 \let\textls\undefined
1675 \let\lslig\undefined
1676 }
1677 </plain>

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

1678 \DeclareRobustCommand\lslig[1]{%
1679   {\MT@ifdefined@c@TF\MT@curr@ls{%
1680     \escapechar\m@ne
1681     \MT@get@ls@basefont
1682     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1683     \kern\MT@outer@kern
1684     \font@name #1%
1685     \kern\MT@outer@kern%
1686   }}{#1}}%
1687 }

\MT@ls@basefont    pdfTEX 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.

1688 \def\MT@get@ls@basefont{%
1689   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1690   \expandafter\ifx\MT@ls@basefont\relax
1691     \MT@exp@two@c\MT@gl@et\MT@ls@basefont\font@name
1692   \else
1693 <debug>\MT@din@fo@n1{1}{... fixing base font}%
1694     \MT@exp@two@c\let\font@name\MT@ls@basefont
1695   \fi
1696 }

```

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

```

1697 \def\MT@set@lsbasefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
1698 \def\MT@set@tr@zero{%
1699   <debug>\MT@info@nl{1}{... zero tracking}%
1700   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1701   \expandafter\ifx\MT@ls@basefont\relax \else
1702   <debug>\MT@info@nl{1}{... fixing base font}%
1703   \aftergroup\MT@set@lsbasefont
1704   \fi
1705 }
```

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

```

1706 <*package>
1707 \MT@requires@pdftex7{
1708   \def\MT@tr@noligatures{%
1709     \ifx\MT@tr@ligatures\empty
1710       \MT@noligatures@\MT@lsfont\@undefined
1711     \else
1712       \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1713     \fi
1714   }
1715 }{
1716   \def\MT@tr@noligatures{%
1717     \MT@warning@nl{%
1718       Disabling selected ligatures is only possible since\MessageBreak
1719       pdftex 1.40.4. Disabling all ligatures instead}%
1720     \MT@glet\MT@tr@noligatures\relax
1721   }
1722 }
```

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

```
1723 \newskip\MT@outer@space
```

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

```

1724 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
1725   <debug>\MT@info@nl2{... orig. space: \the\fontdimen2\MT@lsfont,
1726   <debug>   \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
1727   <debug>   \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
1728   \let\MT@temp\empty
1729   \MT@tr@set@space@{#1}{#4}{2}\empty
1730   \MT@tr@set@space@{#2}{#5}{3}\@plus
1731   \MT@tr@set@space@{#3}{#6}{4}\@minus
1732   \MT@glet\nc{\MT@outer@space\expandafter\string\font@name}\MT@temp
1733   <debug>\MT@info@nl2{... inner space: \the\fontdimen2\MT@lsfont,
1734   <debug>   \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
1735   <debug>\MT@info@nl2{... outer space: \MT@temp}%
1736 }
```

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

```

1737 \def\MT@tr@set@space@#1#2#3#4{%
1738   \MT@ifempty{#2}{%
1739     \MT@ifempty{#1}{%
1740       \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
1741     }{%
1742       \MT@tr@set@space@@{#1}{#3}{1000}%
1743       \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1744       \fontdimen#3\MT@lsfont=\@tempdima

```

```

1745 }%
1746 }{%
1747 \MT@tr@set@space@{#2}{#3}{2000}%
1748 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1749 \MT@ifempty{#1}\relax{%
1750 \MT@tr@set@space@{#1}{#3}{1000}%
1751 \fontdimen#3\MT@lsfont=\@tempdima
1752 }%
1753 }%
1754 }

```

`\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.

```

1755 \def\MT@tr@set@space@#1#2#3{%
1756 \MT@test@ast#1*\@nil{%
1757 \MT@ifdefined@c@TF\MT@tr@unit@
1758 {\edef\@tempb{#1}\MT@scale@to@em}
1759 {\@tempcntb=#1\relax}%
1760 \@tempdima=\dimexpr \dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
1761 -\fontdimen#2\MT@lsfont\relax

```

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

```

1762 \ifnum#2=\tw@
1763 \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
1764 \fi
1765 \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
1766 }{%
1767 \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
1768 \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
1769 }%
1770 <debug>\MT@edinfo{n13{... : font dimen #2 (#1): \the\@tempdima}%
1771 }

```

`\MT@tr@outer@` 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).

```

1772 \def\MT@tr@outer@l{%
1773 \ifhmode
1774 \ifdim\lastskip>5sp
1775 \edef\x{\the\lastskip minus 0pt}%
1776 \setbox\z@\hbox{\MT@outer@space=\x}%
1777 \ifdim\wd\z@>\z@
1778 <debug>\MT@edinfo2{[[[ adjusting pre space: \the\MT@outer@space}%
1779 \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

1780 \let\MT@ls@outer@k\relax
1781 \else

```

The `ragged2e` package sets `\spaceskip` without glue.

```

1782 \ifdim\lastskip=%
1783 \ifnum\spacefactor<2000
1784 \spaceskip
1785 \else
1786 \ifdim\xspaceskip=\z@
1787 \dimexpr\spaceskip+\fontdimen7\font@name\relax
1788 \else
1789 \xspaceskip
1790 \fi
1791 \fi
1792 <debug>\MT@edinfo2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
1793 \unskip \hskip\MT@outer@space\relax

```

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

```

\MT@tr@outer@next
\MT@tr@outer@
\MT@tr@outer@r@ 1800 \def\MT@tr@outer@r{%
                  1801   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
                  1802 }
                  1803 \def\MT@tr@outer@r@{%
                  1804   \def\MT@temp*{}%

```

```
1805 \ifmmode \else
```

```

1806 \ifnum\currentgroup=10 \else
1807 \def\MT@temp*##1{\ifhmode\skip\MT@outer@space
1808 (debug)\MT@info2{}} adjusting post space (1): \the\MT@outer@space}%
1809 \fi}%
1810 \ifcat\egroup\noexpand\MT@tr@outer@next
1811 \ifhmode\unlign\fi\egroup
1812 \MT@set@curr@ok \MT@set@curr@os
1813 \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=}%
1814 \else

```

```

1815     \ifx\maybe@ic\MT@tr@outer@next
1816     \MT@set@curr@ok \MT@set@curr@os
1817     \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=}
1818     \else

```

```

1819 \ifx\check@icr\MT@r@outer@next
1820 \def\MT@temp*{\aftergroup\MT@r@outer@r\check@icr\let\MT@temp=%}
1821 \else
1822 \ifx@sptoken\MT@r@outer@next
1823 \def\MT@temp* {\ifhmode\hskip\MT@outer@space
1824 (debug)\MT@info2[{}]] adjusting post spaces (2): \the\MT@outer@space}%
1825 \fi}%
1826 \else
1827 \ifx-\MT@r@outer@next
1828 \def\MT@temp*{-{\nobreak\hskip\MT@outer@space
1829 (debug)\MT@info2[{}]] adjusting post spaces (3): \the\MT@outer@space}%
1830 }%
1831 \else
1832 \ifx\ \MT@r@outer@next \else
1833 \ifx\space\MT@r@outer@next \else
1834 \ifx\@xobeysp\MT@r@outer@next \else

```

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

```

1835             \def\MT@temp*{\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k
1836 <debug>\MT@edinfo2{--- adjusting post kern: \the\MT@outer@kern}%
1837             \fi}%
1838             \let\MT@tr@outer@next\relax
1839 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
1840 \MT@temp*%
1841 }

```

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

```

\MT@tr@outer@icr@ 1842 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
1843 \def\MT@tr@outer@icr@{%
1844 \let\@let@token=\MT@tr@outer@next
1845 \maybe@ic@
1846 }

```

For older pdfTeX versions, throw an error.

```

1847 }{
1848 \DeclareRobustCommand\lsstyle{%
1849 \MT@error{Letterspacing only works with pdftex version 1.40\MessageBreak
1850 or newer}{Upgrade pdftex, or use the `soul' package instead.}%
1851 \MT@glet\lsstyle\relax
1852 }
1853 }

```

And for luaTeX, too.

```

1854 <*lua>
1855 \MT@requires@luatex{
1856 \DeclareRobustCommand\lsstyle{%
1857 \MT@error{Letterspacing currently doesn't work with luatex}
1858 {Run pdftex, or use the `soul' package instead.}%
1859 \MT@glet\lsstyle\relax
1860 }
1861 }\relax
1862 </lua>
1863 </package>

```

\textls This command may be used like the other text commands. The starred version removes kerning on the sides. The optional argument changes the letterspacing factor.

```

1864 \DeclareRobustCommand\textls{%
1865 \ifstar{\let\MT@ls@adjust@\MT@ls@adjust@empty\MT@textls}%
1866 {\let\MT@ls@adjust@\MT@ls@adjust@relax\MT@textls}%
1867 }

```

\MT@textls This is now almost L^AT_EX's \DeclareTextFontCommand, with the difference that we adjust the outer spacing and kerning also for \lsstyle, while L^AT_EX's text switches don't bother about italic correction.

\MT@letterspace@

```

1868 \newcommand\MT@textls[2][{}]{%
1869 \ifmmode
1870 \nfss@text{\MT@ls@set@ls{#1}\lsstyle#2}%
1871 \else
1872 \hmode@bgroup
1873 \MT@ls@set@ls{#1}%
1874 \lsstyle #2%
1875 \expandafter
1876 \egroup
1877 \fi
1878 }

```

\MT@ls@adjust Set current letterspacing amount and outer kerning. This has to be done inside the

\MT@ls@adjust@empty

\MT@ls@adjust@relax

\MT@ls@set@ls

same group as the letterspacing command.

```

1879 \def\MT@ls@adjust@empty{\let\MT@ls@adjust\@empty}
1880 \def\MT@ls@adjust@relax{\let\MT@ls@adjust\relax}
1881 \def\MT@ls@set@ls#1{%
1882   \MT@ifempty{#1}%
1883   {\let\MT@letterspace@\undefined}%
1884   {\KV@sp@def\MT@letterspace@{#1}%
1885     \MT@ls@too@large\MT@letterspace@}%
1886   \MT@ls@adjust@
1887 }
```

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

```

1888 \def\MT@ls@too@large#1{%
1889   \ifnum#1>\MT@tr@max
1890     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
1891     \let#1\MT@tr@max
1892   \else
1893     \ifnum#1<\MT@tr@min
1894       \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
1895       \let#1\MT@tr@min
1896     \fi
1897   \fi
1898 }
```

`\MT@outer@kern` This dimen is used for the starred version of `\textls`, for `\lslig` and for adjusted
`\MT@tr@set@okern` outer kerning.

```

1899 \newdimen\MT@outer@kern
1900 <package>
1901 \def\MT@tr@set@okern#1,#2,{%
1902   \let\MT@temp\@empty
1903   \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
1904   \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
1905   \MT@glet@nc{\MT@outer@kern\expandafter\string\font@name}\MT@temp
1906   <debug>\MT@diminfo#12{... outer kerning: (#1,#2)
1907   <debug>                = \@nameuse{\MT@outer@kern\expandafter\string\font@name}}%
1908 }
```

`\MT@tr@set@okern@`

```

1909 \def\MT@tr@set@okern@#1{%
1910   \MT@test@ast#1*\@nil{%
1911     \MT@ifdefined@c@TF\MT@tr@unit@
1912     {\edef\@tempb{#1}\MT@scale@to@em}
1913     {\@tempcntb=#1\relax}%
1914     \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
1915   }{%
1916     \MT@ifempty\@tempa{\let\@tempa\@m}\relax
1917     \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
1918     * \fontdimen6\MT@lsfont/2000\relax
1919   }%
1920   \advance\@tempdima -\dimexpr \MT@letterspace@ sp
1921   * \fontdimen6\MT@lsfont/2000\relax
1922   \edef\MT@temp{\MT@temp{\the\@tempdima}}%
1923 }
1924 </package>
```

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

```

1925 \def\MT@ls@outer@k{\ifhmode\kern\MT@outer@kern\relax\fi}
1926 <package>
```

14.2.6 Disabling ligatures

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

```

1927 \MT@requires@pdftex5{
1928 \def\MT@noligatures{%
1929   \MT@dotrue
1930   \let\@tempa\MT@n\setname
1931   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1932     \MT@ifdefined@n@TF{MT@checklist@##1}%
1933     {\csname MT@checklist@##1\endcsname}%
1934     {\MT@checklist@{##1}}%
1935     {n\}%
1936   }%
1937   \ifMT@do
1938     \MT@noligatures@\MT@font\MT@n\ligatures
1939   \fi
1940 }
```

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

```

1941 \def\MT@noligatures@#1#2{%
1942   \MT@ifdefined@c@TF#2{%
```

Early MiK_{TeX} versions (before 2.5.2579) didn't know `\tagcode`.

```

1943   \MT@ifdefined@c@TF\tagcode{%
```

No 'inputenc' key.

```

1944   \let\MT@warn@maybe@inputenc\empty
1945   \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
1946   \MT@map@clist@c#2{%
1947     \KV@sp@def\@tempa{##1}\MT@get@slot
1948     \ifnum\MT@char>\m@ne \tagcode#1\MT@char=\m@ne \fi}%
1949   \MT@vinfo{... Disabling ligatures for characters: #2}%
1950 }{%
1951   \pdfnoligatures#1%
1952   \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
1953     know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
1954     the font instead}%
1955 }%
1956 }{%
1957   \pdfnoligatures#1%
1958   \MT@vinfo{... Disabling ligatures}%
1959 }%
1960 }
1961 }\relax
```

14.2.7 Loading the configuration

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

```

1962 \def\MT@load@list#1{%
1963   \edef\@tempa{#1}%
1964   \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa @load}%
1965   \MT@ifstreq\@tempa\@tempb{%
1966     \MT@error{\@nameuse{MT@abbr@\MT@feat} list `@\@tempa' cannot load itself}{}%
1967   }{%
1968     \ifx\@tempb\relax \else
1969       \MT@ifdefined@n@TF{MT@MT@feat @c@\@tempb}{%
1970         \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list `@\@tempb'}%
1971         \begingroup
1972         \MT@load@list\@tempb
1973         \endgroup
1974       \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list
```

```

1975      \noexpand\MessageBreak` \@tempb' }%
1976      \MT@let@cn\@tempc{MT@MT@feat @c\@tempb}%
1977      \expandafter\MT@set@codes\@tempc,\relax,%
1978      }{%
1979      \MT@error{\@nameuse{MT@abbr@MT@feat} list ` \@tempb' undefined.\MessageBreak
1980              Cannot load it from list ` \@tempa' }{%
1981      }%
1982      \fi
1983      }%
1984      }

```

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

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

```

1985 \let\MT@file@list\empty
1986 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

1987 \MT@in@clist{#1}\MT@file@list
1988 \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.

```

1989 \MT@begin@catcodes
1990 \let\MT@begin@catcodes\relax
1991 \let\MT@end@catcodes\relax
1992 \InputIfFileExists{mt-#1.cfg}{%
1993 \edef\MT@curr@file{mt-#1.cfg}%
1994 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1995 \MT@xadd\MT@file@list{#1,}%
1996 }{%
1997 \MT@get@basefamily#1\@empty\@empty\@empty\@nil
1998 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1999 \ifMT@inlist@
2000 \MT@xadd\MT@file@list{#1,}%
2001 \else
2002 \InputIfFileExists{mt-\@tempa.cfg}{%
2003 \edef\MT@curr@file{mt-\@tempa.cfg}%
2004 \MT@vinfo{... Loading configuration file \MT@curr@file}%
2005 \MT@xadd\MT@file@list{\@tempa,#1,}%
2006 }{%
2007 \MT@vinfo{... No configuration file mt-#1.cfg}%
2008 \MT@xadd\MT@file@list{#1,}%
2009 }%
2010 \fi
2011 }%
2012 \endgroup
2013 \fi
2014 }

```

`\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 L^AT_EX 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.

```

2015 \def\MT@cfg@catcodes{%
2016 \makeatletter
2017 \catcode`\^7%
2018 \catcode`\ 9%

```



```

2019 \catcode\^I9%
2020 \catcode\^M9%
2021 \catcode\\\z@
2022 \catcode\{\@ne
2023 \catcode\}\@tw@
2024 \catcode\#6%
2025 \catcode\%14%
2026 \MT@map@tlist@n
2027 {\!\"$&'(\)\*+,\-\.\/\:\;\<=\>\?[\]\_-\`|/~}%
2028 \makeother
2029 }

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

2030 \def\MT@begin@catcodes{%
2031 \begingroup
2032 \MT@cfg@catcodes
2033 }

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

2034 \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.

2035 \def\MT@get@basefamily#1#2#3#4\@nil{%
2036 \ifx\@empty#4%
2037 \def\@tempa{#1#2#3}%
2038 \else
2039 \let\@tempa\@empty
2040 \edef\@tempb{#1#2#3#4}%
2041 \expandafter\MT@get@basefamily@\@tempb\@nil
2042 \fi
2043 }

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

2044 \def\MT@get@basefamily@#1#2\@nil{%
2045 \edef\@tempa{\@tempa#1}%
2046 \ifx\#2\expandafter\@gobble\else\expandafter\@firstofone\fi
2047 {\MT@in@tlist{#2}\MT@variants
2048 \ifMT@inlist\else\MT@get@basefamily@#2\@nil\fi}%
2049 }

\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@ 2050 \def\MT@get@listname#1{%
2051 \MT@info{n1}{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
2052 \let\MT@listname\undefined
2053 \def\@tempb{#1}%
2054 \MT@map@tlist@c\MT@try@order\MT@get@listname@
2055 }
2056 \def\MT@get@listname@#1{%
2057 \expandafter\MT@next@listname#1%
2058 \ifx\MT@listname\undefined \else
2059 \expandafter\MT@tlist@break
2060 \fi

```

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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

2061 }

\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.

```
2062 \def\MT@try@order{%
2063   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2064   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2065 }
```

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

```
2066 \def\MT@next@listname#1#2#3#4{%
2067   \edef\@tempa{\MT@encoding
2068     /\ifnum#1=\@ne \MT@family\fi
2069     /\ifnum#2=\@ne \MT@series\fi
2070     /\ifnum#3=\@ne \MT@shape\fi
2071     /\ifnum#4=\@ne *\fi
2072     \MT@context}%
2073   <debug>\MT@info@n1{1}{trying \@tempa}%
2074   \MT@ifdefined@n@TF{\MT@\@tempb \@tempa}{%
2075     \MT@next@listname@#4%
2076   }{%
```

Also try with an alias family.

```
2077   \ifnum#1=\@ne
2078     \ifx\MT@familyalias\@empty \else
2079       \edef\@tempa{\MT@encoding
2080         /\MT@familyalias
2081         /\ifnum#2=\@ne \MT@series\fi
2082         /\ifnum#3=\@ne \MT@shape\fi
2083         /\ifnum#4=\@ne *\fi
2084         \MT@context}%
2085     <debug>\MT@info@n1{1}{(alias) \@tempa}%
2086     \MT@ifdefined@n@T{\MT@\@tempb \@tempa}{%
2087       \MT@next@listname@#4%
2088     }%
2089   \fi
2090   \fi
2091 }%
2092 }
```

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

```
2093 \def\MT@next@listname@#1{%
2094   \ifnum#1=\@ne
2095     \MT@exp@cs\MT@in@rlist{\MT@\@tempb \@tempa @sizes}%
2096     \ifMT@inlist@
2097       \let\MT@listname\MT@size@name
2098     \fi
2099   \else
2100     \MT@let@cn\MT@listname{\MT@\@tempb \@tempa}%
2101   \fi
```

2102 }

\MT@if@list@exists

```

\MT@context 2103 \def\MT@if@list@exists{%
2104   \MT@let@cn\MT@context{MT@\MT@feat @context}%
2105   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
2106   \MT@get@listname{\MT@feat @c}%
2107   \MT@ifdefined@c@TF\MT@listname{%
2108     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
2109     \ifMT@nonselected
2110       \MT@vinfo{... Applying non-selected expansion (list '\MT@listname')}%
2111     \else
2112       \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list '\MT@listname'}%
2113     \fi
2114     \@firstoftwo
2115   }%

```

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

```
2116   \MT@let@nc{MT@\MT@feat @c@name}\@empty
```

Don't warn if selected=false.

```

2117   \ifMT@nonselected
2118     \MT@vinfo{... Applying non-selected expansion (no list)}%
2119   \else

```

Tracking doesn't require a list, either.

```

2120     \MT@ifstreq\MT@feat{tr}\relax{%
2121       \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
2122         for font\MessageBreak'\MT@font'%
2123         \ifx\MT@context\@empty\else\space(context: '\MT@context')\fi.
2124       Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
2125     }%
2126   \fi
2127   \@secondoftwo
2128 }%
2129 }

```

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

```

\MT@context 2130 \def\MT@get@inh@list{%
2131   \let\MT@context\@empty
2132   \MT@get@listname{\MT@feat @inh}%
2133   \MT@ifdefined@c@TF\MT@listname{%
2134     \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
2135   <debug>\MT@dinfo@n{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
2136   <debug>                                '\MT@listname'}%
2137   \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%

```

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

```

2138   \ifx\@tempc\@empty \else
2139   <debug>\MT@dinfo@n{1}{parsing inheritance list ...}%

```

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

```

2140     \begingroup
2141     \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak'\MT@listname'}%
2142     \MT@set@inputenc{inh}%
2143     \expandafter\MT@inh@do\@tempc,\relax,%
2144     \MT@glet@nc{MT@\MT@feat @inh@\MT@listname}\@empty
2145   \endgroup
2146   \fi
2147 }%
2148   \MT@let@nc{MT@\MT@feat @inh@name}\@undefined

```

```

2149 }%
2150 }

```

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@ 2151 \def\MT@get@slot{%
2152   \escapechar~\
2153   \let\MT@char@mone
2154   \MT@noresttrue

```

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

```

2155 \MT@toks=\expandafter{\@tempa}%

```

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

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

```

2156 \expandafter\MT@is@letter\@tempa\relax\relax
2157 \ifnum\MT@char@ < \z@

```

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

```

2158 \MT@exp@two@c\MT@is@active\string\@tempa\@nil

```

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

If `\<encoding>\<command>` (that's *one* command) is defined, we try to extract the slot number.

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

```

2159 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
2160 \MT@is@symbol

```

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

```

2161 {\expandafter\MT@is@composite\@tempa\relax\relax}%
2162 \ifnum\MT@char@ < \z@

```

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

```

2163 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2164 \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2165 \fi
2166 \fi

2167 \let\MT@char\MT@char@
2168 \ifnum\MT@char@ < \z@
2169 \MT@warn@unknown
2170 \else

```

If the user has specified something like ‘fi’, or wanted to define a number but forgot to use three digits, we’ll have something left of the string. In this case, we issue a warning and forget the complete string.

```

2171     \ifMT@norest \else
2172         \MT@warn@rest
2173         \let\MT@char\m@ne
2174     \fi
2175 \fi
2176 \escapechar\m@ne
2177 }

\ifMT@norest    Test whether all of the string has been used up.
2178 \newif\ifMT@norest

\MT@is@letter    Input is a letter, a character or a number.
2179 \def\MT@is@letter#1#2\relax{%
2180     \ifcat a\noexpand#1\relax
2181         \edef\MT@char@{\number`#1}%
2182         \ifx\#2\%
2183             (debug)\MT@info@n1{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2184         \else
2185             \MT@norestfalse
2186         \fi
2187     \else
2188         \ifcat !\noexpand#1\relax
2189             \edef\MT@char@{\number`#1}%
2190             (debug)\MT@info@n1{3}{> `the\MT@toks' is a character (\MT@char@)}%
2191             \ifx\#2\%
2192                 \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
2193             \else
2194                 \MT@norestfalse
2195                 \expandafter\MT@is@number#1#2\relax\relax
2196             \fi
2197         \fi
2198     \fi
2199 }

\MT@is@number    Numbers may be specified as a three-digit decimal number (029), as a hexadecimal
                  number (prefixed with ": "1D) or as a octal number (prefixed with ': '35). They
                  must consist of at least three characters (including the prefix), that is, "F is not
                  permitted.
2200 \def\MT@is@number#1#2#3\relax{%
2201     \ifx\relax#3\relax \else
2202         \ifx\relax#2\relax \else
2203             \MT@noresttrue
2204             \if#1"\relax
2205                 \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2206                 (debug)\MT@info@n1{3}{> ... a hexadecimal number: \MT@char@}%
2207             \else
2208                 \if#1'\relax
2209                     \def\MT@char@{\number#1#2#3}%
2210                 (debug)\MT@info@n1{3}{> ... an octal number: \MT@char@}%
2211             \else
2212                 \MT@ifint{#1#2#3}{%
2213                     \def\MT@char@{\number#1#2#3}%
2214                 (debug)\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
2215                 }\MT@norestfalse
2216             \fi
2217         \fi
2218     \ifnum\MT@char@ > \cc1v

```

```

2219      \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2220      \let\MT@char@m@ne
2221      \fi
2222      \fi
2223      \fi
2224  }

```

`\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.

```

2225 \def\MT@is@active#1#2\@nil{%
2226   \ifnum\catcode`#1 = \active
2227     \begingroup
2228     \set@display@protect
2229     \let\IeC\@firstofone
2230     \let\@inpenc@undefined@MT@undefined@char

```

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

```

2231   \def\UTFviii@defined##1{\ifx ##1\relax
2232     \MT@undefined@char{utf8}\else\expandafter ##1\fi}%

```

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

```

2233   \MT@ifdefined@c@T\PrerenderUnicode
2234   {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
2235   \edef\x{\endgroup
2236     \def\noexpand\@tempa{\@tempa}%

```

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

```

2237     \MT@toks={\the\MT@toks\space(= \@tempa)}%
2238     }%
2239     \x
2240     \fi
2241  }

```

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

```

2242 \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.

```

2243 \def\MT@is@symbol{%
2244   \expandafter\def\expandafter\MT@char\expandafter
2245     {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2246   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2247     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2248   \ifnum\MT@char@ < \z@

```

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

```

2249     \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2250     \fi

```

2251 }

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

```

\MT@charstring 2252 \begingroup
2253   \catcode`\=/=\z@
2254   /MT@map@tlist@n{/CHAR}/@makeoether
2255   /lowercase{%
2256     /def/x{/endgroup
2257       /def/MT@charstring{CHAR}%
2258       /def/MT@is@char#1CHAR"##2##3##4/relax{%
2259         /ifx/relax#1/relax
2260         /if##3\relax
2261           /edef/MT@char@{/number"##2}%
2262           /MT@ifstreq/MT@charstring{##3##4}/relax/MT@noestfalse
2263         /else
2264           /edef/MT@char@{/number"##2##3}%
2265           /MT@ifstreq/MT@charstring{##4}/relax/MT@noestfalse
2266         /fi
2267 (debug) /MT@dinfo@n1{3}{> `the/MT@toks' is a \char (/MT@char@)}%
2268       /fi
2269     }%
2270   }%
2271 }
2272 /x

```

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

```

2273 \def\MT@is@composite#1#2\relax{%
2274   \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`.

```

2275   \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2276     \string\csname\MT@encoding\endcsname
2277     \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
2278   \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2279   \fi
2280 }

```

[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 2281 \def\MT@set@list@name{%
2282   \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
2283   \@nameuse{MT@\MT@feat @c@name}}}%
2284 }

\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.

2285 \def\MT@warn@ascii{%
2286   \MT@warning@nl{Character \the\MT@toks' (= \MT@char@)
2287   is outside of ASCII range.\MessageBreak
2288   You must load the ‘inputenc’ package before using\MessageBreak
2289   8-bit characters in \MT@curr@list@name}%
2290 }

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

2291 \def\MT@warn@number@too@large#1{%
2292   \MT@warning@nl{%
2293     Number #1 in encoding ‘\MT@encoding’ too large!\MessageBreak
2294     Ignoring it in \MT@curr@list@name}%
2295 }

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

2296 \def\MT@warn@rest{%
2297   \MT@warning@nl{%
2298     Unknown slot number of character\MessageBreak\the\MT@toks'
2299     \MT@warn@maybe@inputenc\MessageBreak
2300     in font encoding ‘\MT@encoding’.\MessageBreak
2301     Make sure it's a single character\MessageBreak
2302     (or a number) in \MT@curr@list@name}%
2303 }

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

2304 \def\MT@warn@unknown{%
2305   \MT@warning@nl{%
2306     Unknown slot number of character\MessageBreak\the\MT@toks'
2307     \MT@warn@maybe@inputenc\MessageBreak
2308     in font encoding ‘\MT@encoding’ in \MT@curr@list@name}%
2309 }

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

2310 \def\MT@warn@maybe@inputenc{%
2311   \MT@ifdefined@n@T
2312   {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}%
2313   { (input encoding \@nameuse
2314   {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2315 }

```

14.2.9 Hook into L^AT_EX’s font selection

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

In contrast to the pdfcpot package, 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, pi font (loaded by hyperref), tipa, and probably many more. Furthermore, we had to include a hack for the IEEEtran class which loads all fonts in the class file itself (to fine tune inter-word spacing), 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 2316 \let\MT@font@list\empty
2317 \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`.

```
2318 </package>
2319 <plain>\MT@requires@latex2{
2320 \MT@addto@setup{%
```

`\MT@orig@pickupfont` 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 expansion. (Tracking doesn't really work for other reasons.) Like us, CJK redefines `\pickup@font`.

```
2321 \ifpackageloaded{CJK}{%
2322 \ifpackageafter{CJK}{2006/10/17}% 4.7.0
2323 {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}}}%
2324 {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}%
2325 \g@addto@macro\MT@orig@pickupfont
2326 {\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).

```

2327 \ifpackageloaded{CJKutf8}%
2328 {\ifpackageolder{CJKutf8}{2008/05/22}% 4.8.0
2329 {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}%
2330 {\@firstoftwo}}%
2331 {\@firstoftwo}%
2332 {\g@addto@macro\MT@orig@pickupfont{%
2333 {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2334 \define@newfont\else\xdef\font@name{%
2335 \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
2336 {\g@addto@macro\MT@orig@pickupfont{%
2337 {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2338 \define@newfont\def\CJK@temp{v}%
2339 \ifx\CJK@temp\CJK@plane
2340 \expandafter\ifx\csname CJK@cmap@\f@family\CJK@plane\endcsname\relax
2341 \else\csname CJK@cmap@\f@family\CJK@plane\endcsname\fi
2342 \else \CJK@addcmap\CJK@plane \fi
2343 \else\xdef\font@name{%
2344 \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
2345 }%
2346 \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}%
2347 }%

```

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

```

2348 \ifx\pickup@font\MT@orig@pickupfont \else
2349 \MT@warning@nl{%
2350 Command \string\pickup@font\space is not defined as expected.%
2351 \MessageBreak Patching it anyway. Some things may break%
2352 }%
2353 \MessageBreak Double-check whether micro-typography is indeed%
2354 \MessageBreak applied to the document.%
2355 \MessageBreak (Hint: Turn on `verbose' mode)%
2356 }%
2357 }%
2358 \fi

```

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

```

2359 \g@addto@macro\pickup@font{\begingroup}%

```

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

```

2360 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
2361 \g@addto@macro\pickup@font{%
2362 \escapechar\m@ne
2363 }%
2364 \global\MT@inannottrue
2365 \MT@gl@et\MT@pdf@annot\@empty
2366 \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.

```

2367 \MT@let@cn\MT@font\MT@subst\expandafter\string\font@name}%
2368 \ifx\MT@font\relax
2369 \let\MT@font\font@name
2370 \else
2371 \ifx\MT@font\font@name \else
2372 \MT@addto@annot{= substituted with \MT@font}%
2373 \MT@register@subst@font
2374 \fi

```

```

2375     \fi
2376     \MT@setupfont
2377 \end{package}
2378 \letterspace \MT@tracking
2379 \endgroup
2380 }%
2381 \end{package}

\MT@pickupfont Remember the patched command for later.
2382 \let\MT@pickupfont\pickupfont

\do@subst@correction Additionally, we hook into \do@subst@correction, which is called if a substitution
has taken place, to record the name of the ersatz font. Unfortunately, this will only
work for one-level substitutions. We have to remember the substitute for the rest of
the document, not just for the first time it is called, since we need it every time a
font is letterspaced.
2383 \g@addto@macro\do@subst@correction
2384 { \edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2385   \MT@glet\nc{MT@subst@expandafter\string\font@name}\MT@font}%

\add@accent Inside \add@accent, we have to disable microtype's setup, since the grouping in
\MT@orig@add@accent the patched \pickupfont would break the accent if different fonts are used for
the base character and the accent. Fortunately, LATEX takes care that the fonts used
for the \accent are already set up, so that we cannot be overlooking them.
2386 \let\MT@orig@add@accent\add@accent
2387 \def\add@accent#1#2{%
2388   \let\pickupfont\MT@orig@pickupfont
2389   \MT@orig@add@accent{#1}{#2}%
2390   \let\pickupfont\MT@pickupfont
2391 }%
2392 \end{package}
2393 }
2394 \plain\relax
2395 \end{package}

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.

\MT@check@font Check whether we've already seen the current font.
2396 \def\MT@check@font{\MT@expone@n\MT@in@cllist\MT@font\MT@font@list}

\MT@register@subst@font Register the substituted font.
2397 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}

\MT@register@font Register the current font.
2398 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}

```

14.2.10 Context-sensitive setup

Here are the variants for context-sensitive setup.

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

```

2399 \let\MT@active@features\@empty

```

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

```

2400 \def\MT@check@font@cx{%
2401   \MT@if@true
2402   \MT@map@clist@c\MT@active@features{%
2403     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
2404     \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
2405     \ifMT@inlist@
2406       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
2407     \else
2408       \MT@if@false
2409     \fi
2410   }%
2411   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2412 }

```

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

```

2413 \def\MT@register@subst@font@cx{%
2414   \MT@map@clist@c\MT@active@features{%
2415     \MT@exp@cs\MT@xadd
2416     {MT@##1\csname MT@##1@context\endcsname font@list}%
2417     {\font@name,}%
2418   }%
2419 }

```

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

```

2420 \def\MT@register@font@cx{%
2421   \MT@map@clist@c\MT@active@features{%
2422     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
2423     \MT@exp@cs\MT@xadd
2424     {MT@##1\csname MT@##1@context\endcsname font@list}%
2425     {\MT@font,}%
2426     \def\@tempa{##1}%
2427     \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
2428   \fi
2429   }%
2430 }

```

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

```

2431 \def\MT@maybe@rem@from@list#1{%
2432   \MT@if@streq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
2433     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
2434     \MT@font \csname MT@\@tempa @#1font@list\endcsname
2435   }%
2436 }

```

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

Inside the preamble, it shouldn't actually do anything but remember it for later.

```

2437 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
2438 \MT@addto@setup{%
2439   \DeclareRobustCommand\microtypecontext[1]{%
2440     \MT@setup@contexts
2441     \let\MT@reset@context\relax
2442     \setkeys{MTC}{#1}%
2443     \selectfont
2444     \MT@reset@context
2445   }%
2446 }

```

`\textmicrotypecontext` This is just a wrapper around `\microtypecontext`.

```

2447 \DeclareRobustCommand\textmicrotypecontext[2]{\microtypecontext{#1}#2}

```

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

```

2448 \def\MT@reset@context{%
2449   \MT@vinfo{<<< Resetting contexts\on@line
2450   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
2451   <debug>           /\MT@tr@context/\MT@kn@context/\MT@sp@context
2452   }%
2453   \selectfont
2454 }
```

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

```

2455 \def\MT@setup@contexts{%
2456   \MT@map@clist@c\MT@active@features
2457   {\MT@gl@et@nc{MT@##1@font@list}\MT@font@list}%
2458   \MT@gl@et\MT@check@font\MT@check@font@cx
2459   \MT@gl@et\MT@register@font\MT@register@font@cx
2460   \MT@gl@et\MT@register@subst@font\MT@register@subst@font@cx
2461   \MT@gl@et\MT@setup@contexts\relax
2462 }
```

Define context keys.

```

2463 \MT@map@clist@c\MT@features@long{%
2464   \define@key{MTC}{#1}[]{}%
2465   \edef\@tempb{\@nameuse{MT@rbba#1}}%
2466   \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2467   \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).

```

2468   \MT@ifempty{#1}{\def\MT@val{0}}{\def\MT@val{##1}}%
2469   \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
2470   <debug> \MT@dinfo{1}{>>> no change of #1 context: `'\MT@val'}%
2471   \else
2472     \MT@vinfo{>>> Changing #1 context to `'\MT@val'\MessageBreak\on@line
2473     <debug>           \space(previous: `'\@nameuse{MT@\@tempb @context}')}%
2474     }%
2475   \def\MT@reset@context{\aftergroup\MT@reset@context@}%
```

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

```

2476   \MT@gl@et@nn{MT@reset@\@tempb @codes}\MT@reset@\@tempb @codes}%
```

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

```

2477   \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
2478   \MT@val \csname MT@\@tempb @doc@contexts\endcsname
2479   \ifMT@inlist@ \else
2480     \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{\MT@val}%
2481     <debug> \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
2482     \fi
2483     \MT@edef@n{MT@\@tempb @context}{\MT@val}%
2484     \fi
2485   \fi
2486 }%
2487 }
```

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

```

\MT@ex@context 2488 \MT@exp@one@n\MT@map@clist@n{\MT@features,n1}%
\MT@tr@context 2489 \MT@def@n{MT@#1@context}{@}%
                2490 \MT@def@n{MT@#1@doc@contexts}{\@}%
\MT@sp@context 2491 }
\MT@kn@context 2492 \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`

`\MT@extra@context`

14.3 Configuration

14.3.1 Font sets

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

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

```

2493 \def\DeclareMicrotypeSet{%
2494   \ifstar
2495     \MT@DeclareSetAndUseIt
2496     \MT@DeclareSet
2497 }

\MT@DeclareSet

2498 \newcommand\MT@DeclareSet[3] [] {%
2499   \KV@sp@def\@tempa{#1}%
2500   \MT@ifempty\@tempa{%
2501     \MT@map@clistc\MT@features{\MT@declare@sets{##1}{#2}{#3}}}%
2502   }{%
2503     \MT@map@clistc\@tempa{%
2504       \KV@sp@def\@tempa{##1}%
2505       \MT@ifempty\@tempa\relax{%
2506         \MT@is@feature{set declaration `#2'}{%
2507           \MT@exp@one@n\MT@declare@sets
2508             {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
2509         }%
2510       }%
2511     }%
2512   }%
2513 }

```

`\MT@DeclareSetAndUseIt`

```

2514 \newcommand\MT@DeclareSetAndUseIt[3] [] {%
2515   \MT@DeclareSet[#1]{#2}{#3}%
2516   \UseMicrotypeSet[#1]{#2}%
2517 }

```

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

```

2518 \let\MT@curr@set@name\@empty

```

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

```

2519 \def\MT@declare@sets#1#2#3{%
2520   \KV@sp@def\MT@curr@set@name{#2}%
2521   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
2522     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2523     \MT@glet@nc{MT@#1list@size@\MT@curr@set@name}\@empty
2524   }%
2525   \MT@glet@nc{MT@#1@set@@\MT@curr@set@name}\@empty
2526   <debug>\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2527   \setkeys{MT@#1@set}{#3}%
2528 }

```

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

```

2529 \def\MT@define@set@key@#1#2{%
2530   \define@key{MT@#2@set}{#1} [] {%
2531     \MT@glet@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
2532     \MT@map@clistn{##1}{%

```

```

2533 \KV@sp@def\MT@val{###1}%
2534 \MT@get@highlevel{#1}%

```

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

```

2535 \MT@exp@two@n@g@addto@macro
2536 {\csname MT@#2list@#1@MT@curr@set@name\expandafter\endcsname}%
2537 {\MT@val,}%
2538 }%

```

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

```

2539 \expandafter\g@addto@macro\expandafter\MT@font@sets
2540 \csname MT@#2list@#1@MT@curr@set@name\endcsname
2541 <debug>\MT@info@n1{1}{-- #1: \@nameuse{MT@#2list@#1@MT@curr@set@name}}%
2542 }%
2543 }

```

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

```

2544 \def\MT@get@highlevel#1{%
2545 \expandafter\MT@test@ast\MT@val*\@nil\relax}%

```

And ‘family = *’ will become \familydefault.

```

2546 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
2547 \edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}%

```

In contrast to earlier version, these values will not be expanded immediately but at the end of the preamble.

```

2548 }%
2549 }

```

\MT@test@ast It the last character is an asterisk, execute the second argument, otherwise the first one.

```

2550 \def\MT@test@ast#1*#2\@nil{%
2551 \def\@tempa{#1}%
2552 \MT@ifempty{#2}%
2553 }

```

\MT@font@sets Fully expand the font specification and fix catcodes for all font sets.

```

\MT@fix@font@set 2554 \let\MT@font@sets\empty
2555 \def\MT@fix@font@set#1{%
2556 \xdef#1{#1}%
2557 \global\@onelevel@sanitize#1%
2558 }

```

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

```

2559 \def\MT@define@set@key@size#1{%
2560 \define@key{MT@#1@set}{size}[]{}%
2561 \MT@map@clist@n{##1}{%
2562 \KV@sp@def\MT@val{###1}%
2563 \expandafter\MT@get@range\MT@val--\@nil
2564 \ifx\MT@val\relax \else
2565 \MT@exp@cs\MT@xadd
2566 {MT@#1list@size@MT@curr@set@name}%
2567 {{{\MT@lower}{\MT@upper}\relax}}%
2568 \fi
2569 }%
2570 <debug>\MT@info@n1{1}{-- size: \@nameuse{MT@#1list@size@MT@curr@set@name}}%
2571 }%
2572 }

```

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

`\MT@get@range` Ranges will be stored as triplets of $\{\langle lower\ bound \rangle\}\{\langle upper\ bound \rangle\}\{\langle list\ name \rangle\}$.
`\MT@upper` For simple sizes, the upper boundary is -1 .

```

\MT@lower 2573 \def\MT@get@range#1-#2-#3\@nil{%
2574   \MT@ifempty{#1}{%
2575     \MT@ifempty{#2}{%
2576       \let\MT@val\relax
2577     }{%
2578       \def\MT@lower{0}%
2579       \def\MT@val{#2}%
2580       \MT@get@size
2581       \edef\MT@upper{\MT@val}%
2582     }%
2583   }{%
2584     \def\MT@val{#1}%
2585     \MT@get@size
2586     \ifx\MT@val\relax \else
2587       \edef\MT@lower{\MT@val}%
2588       \MT@ifempty{#2}{%
2589         \MT@ifempty{#3}%
2590         {\def\MT@upper{-1}}%

```

2048 pt is TeX's maximum font size.

```

2591   {\def\MT@upper{2048}}%
2592 }{%
2593   \def\MT@val{#2}%
2594   \MT@get@size
2595   \ifx\MT@val\relax \else
2596     \MT@ifdim\MT@lower>\MT@val{%
2597       \MT@error{%
2598         Invalid size range (\MT@lower\space > \MT@val) in font set
2599         ~\MT@curr@set@name'.\MessageBreak Swapping sizes}}%
2600     \edef\MT@upper{\MT@lower}%
2601     \edef\MT@lower{\MT@val}%
2602   }{%
2603     \edef\MT@upper{\MT@val}%
2604   }%
2605   \MT@ifdim\MT@lower=\MT@upper
2606   {\def\MT@upper{-1}}%
2607   \relax
2608 \fi
2609 }%
2610 \fi
2611 }%
2612 }

```

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

```

2613 \def\MT@get@size{%

```

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

```

2614   \if*\MT@val\relax
2615     \def\@tempa{\normalsize}%
2616   \else
2617     \MT@let@cn\@tempa{\MT@val}%
2618   \fi

```



```
2619 \ifx\@tempa\relax \else
```

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

```
2620 \begingroup
2621 \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
2622 \@tempa\@nil
2623 \fi
```

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

```
2624 \MT@ifdimen\MT@val{%
2625 \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2626 \edef\MT@val{\strip@pt\@tempdima}%
2627 }{%
2628 \MT@warning{Could not parse font size `~\MT@val'\MessageBreak
2629 in font set `~\MT@curr@set@name'}%
2630 \let\MT@val\relax
2631 }%
2632 }
```

\MT@define@set@key@font

```
2633 \def\MT@define@set@key@font#1{%
2634 \define@key{MT@#1@set}{font}[]{%
2635 \MT@glet@nc{MT@#1list@font@MT@curr@set@name}\@empty
2636 \MT@map@clist@n{##1}{%
2637 \KV@esp@def\MT@val{####1}%
2638 \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/}}\relax
2639 \expandafter\MT@get@font\MT@val////\@nil
2640 \MT@exp@two@n@g@addto@macro
2641 {\csname MT@#1list@font@MT@curr@set@name\expandafter\endcsname}%
2642 {\MT@val,}%
2643 }%
2644 \expandafter\g@addto@macro\expandafter\MT@font@sets
2645 \csname MT@#1list@font@MT@curr@set@name\endcsname
2646 (debug)\MT@dinfol{1}{-- font: \@nameuse{MT@#1list@font@MT@curr@set@name}}%
2647 }%
2648 }
```

\MT@get@font Translate any asterisks.

```
2649 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2650 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
2651 \ifx\MT@val\relax\def\MT@val{0}\fi
2652 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2653 \let\MT@val\@tempb
2654 }
```

\MT@get@font@ Helper macro, also used by \MT@get@font@and@size.

```
2655 \def\MT@get@font@#1#2#3#4#5#6{%
2656 \let\@tempb\@empty
2657 \def\MT@temp{#1/#2/#3/#4/#5}%
2658 \MT@get@axis{encoding}{#1}%
2659 \MT@get@axis{family}{#2}%
2660 \MT@get@axis{series}{#3}%
2661 \MT@get@axis{shape}{#4}%
2662 \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
2663 \MT@ifempty{#5}{%
2664 \MT@warn@axis@empty{size}{\string\normalsize}%
2665 \def\MT@val{*}%
2666 }
```

```

2666 }{%
2667   \def\MT@val{#5}%
2668 }%
2669 \MT@get@size
2670 }

```

\MT@get@axis

```

2671 \def\MT@get@axis#1#2{%
2672   \def\MT@val{#2}%
2673   \MT@get@highlevel{#1}%
2674   \MT@ifempty\MT@val{%
2675     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2676     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2677   }\relax
2678   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2679 }

```

\MT@warn@axis@empty

```

2680 \def\MT@warn@axis@empty#1#2{%
2681   \MT@warning{#1 axis is empty in font specification\MessageBreak
2682     ~\MT@temp'. Using ~#2' instead}%
2683 }

```

We can finally assemble all pieces to define \DeclareMicrotypeSet's keys. They are also used for \DisableLigatures.

```

2684 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
2685   \MT@define@set@key@{encoding}{#1}%
2686   \MT@define@set@key@{family}{#1}%
2687   \MT@define@set@key@{series}{#1}%
2688   \MT@define@set@key@{shape}{#1}%
2689   \MT@define@set@key@size{#1}%
2690   \MT@define@set@key@font{#1}%
2691 }

```

\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.

```

2692 \renewcommand*\UseMicrotypeSet[2][]{%
2693   \KV@sp@def\@tempa{#1}%
2694   \MT@ifempty\@tempa{%
2695     \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
2696   }{%
2697     \MT@map@clist@c\@tempa{%
2698       \KV@sp@def\@tempa{##1}%
2699       \MT@ifempty\@tempa\relax{%
2700         \MT@is@feature{activation of set ~#2'}{%
2701           \MT@exp@one@n\MT@use@set
2702             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2703         }%
2704       }%
2705     }}%
2706   }%
2707 }

```

\MT@pr@setname

Only use sets that have been declared.

```

\MT@ex@setname 2708 \def\MT@use@set#1#2{%
\MT@tr@setname 2709   \KV@sp@def\@tempa{#2}%
2710   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@sp@setname 2711     \MT@xdef\MT@#1@setname{\@tempa}%
\MT@kn@setname 2712 }{%
2713   \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2714     \MT@xdef\MT@#1@setname{\@nameuse{MT@default@#1@set}}%
2715   }%

```

```

2716 \MT@error{%
2717   The \nameuse{MT@abbr@#1} set '\@tempa' is undeclared.\MessageBreak
2718   Using set '\nameuse{MT@#1@setname}' instead}{}%
2719 }%
2720 }

```

`\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.

```

2721 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2722   \KV@sp@def\@tempa{#1}%
2723   \MT@ifempty\@tempa{%
2724     \MT@map@clist@c\MT@features{\MT@set@default@set{##1}{#2}}}%
2725   }{%
2726     \MT@map@clist@c\@tempa{%
2727       \KV@sp@def\@tempa{##1}%
2728       \MT@ifempty\@tempa\relax{%
2729         \MT@is@feature{declaration of default set '#2'}{%
2730           \MT@exp@one@n\MT@set@default@set
2731           {\csname MT@rbba@\@tempa\endcsname}{#2}%
2732         }%
2733       }%
2734     }%
2735   }%
2736 }

```

```

\MT@default@pr@set
\MT@default@ex@set 2737 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 2738   \KV@sp@def\@tempa{#2}%
2739   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@default@sp@set 2740   (debug)\MT@info{1}{declaring default \nameuse{MT@abbr@#1} set '\@tempa'}%
\MT@default@kn@set 2741   \MT@xdef@n{MT@default@#1@set}{\@tempa}%
\MT@set@default@set 2742   }{%
2743     \MT@error{%
2744       The \nameuse{MT@abbr@#1} set '\@tempa' is not declared.\MessageBreak
2745       Cannot make it the default set. Using set\MessageBreak `all' instead}{}%
2746     \MT@xdef@n{MT@default@#1@set}{all}%
2747   }%
2748 }

```

14.3.2 Variants and aliases

`\DeclareMicrotypeVariants` Specify suffixes for variants (see `fontname/variants.map`). The starred version appends to the list.

```

\MT@variants
2749 \let\MT@variants\@empty
2750 \def\DeclareMicrotypeVariants{%
2751   \ifstar
2752     \MT@DeclareVariants
2753     {\let\MT@variants\@empty\MT@DeclareVariants}%
2754 }

```

```

\MT@DeclareVariants
2755 \def\MT@DeclareVariants#1{%
2756   \MT@map@clist@n{#1}{%
2757     \KV@sp@def\@tempa{##1}%
2758     \@onelevel@sanitize\@tempa
2759     \xdef\MT@variants{\MT@variants{\@tempa}}%
2760   }%
2761 }

```

`\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.

```

2762 \renewcommand*{\DeclareMicrotypeAlias}[2]{%
2763   \KV@sp@def\@tempa{#1}%
2764   \KV@sp@def\@tempb{#2}%
2765   \onelevel@sanitize\@tempb
2766   \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2767     \MT@warning{Alias font family '\@tempb' will override
2768       alias '\@nameuse{MT@\@tempa @alias}'\MessageBreak
2769       for font family '\@tempa'}}%
2770   \MT@xdef@n{MT@\@tempa @alias}{\@tempb}%

```

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

```

2771   \MT@ifdefined@c@T{MT@family{%
2772     (debug)\MT@info{1}{Activating alias font '\@tempb' for '\MT@family'}%
2773     \MT@gl@et{MT@familyalias\@tempb
2774     }%
2775   }

```

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

```

2776 \def\LoadMicrotypeFile#1{%
2777   \KV@sp@def\@tempa{#1}%
2778   \onelevel@sanitize\@tempa
2779   \MT@exp@one@n{MT@in@clist\@tempa\MT@file@list
2780   \ifMT@inlist@
2781     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2782   \else
2783     \MT@xadd\MT@file@list{\@tempa,}%
2784     \MT@begin@catcodes
2785     \InputIfFileExists{mt-\@tempa.cfg}{%
2786       \edef\MT@curr@file{mt-\@tempa.cfg}%
2787       \MT@vinfo{... Loading configuration file \MT@curr@file}%
2788     }{%
2789       \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2790         does not exist}%
2791     }%
2792     \MT@end@catcodes
2793   \fi
2794 }

```

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 2795 \MT@requires@pdftex5{
2796   \def\DisableLigatures{%
2797     \MT@begin@catcodes
2798     \MT@DisableLigatures
2799   }
2800   \newcommand*\MT@DisableLigatures[2][ ]{%
2801     \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
2802     \xdef\MT@active@features{\MT@active@features,nl}%
2803     \global\MT@noligaturestrue
2804     \MT@declare@sets{nl}{no ligatures}{#2}%
2805     \gdef\MT@nl@setname{no ligatures}%
2806     \MT@end@catcodes
2807   }
2808 }{

```

If pdfTeX is too old, we throw an error.

```
2809 \renewcommand*\DisableLigatures[2] [] {%
2810   \MT@error{Disabling ligatures of a font is only possible\MessageBreak
2811     with pdftex version 1.30 or newer.\MessageBreak
2812     Ignoring \string\DisableLigatures}{Upgrade pdftex.}%
2813 }
2814 }
```

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.

```
2815 \def\DeclareMicrotypeBabelHook#1#2{%
2816   \MT@map@clist@n{#1}{%
2817     \KV@esp@def\@tempa{##1}%
2818     \MT@gdef@n{MT@babel@{\@tempa}{#2}%
2819   }%
2820 }
```

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).

```
2821 \def\SetProtrusion{%
2822   \MT@begin@catcodes
2823   \MT@SetProtrusion
2824 }
```

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

```
\MT@pr@c@name 2825 \newcommand*\MT@SetProtrusion[3] [] {%
```

`\MT@extra@context` 2826 \let\MT@extra@context\empty

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

```
2827 \MT@set@named@keys{MT@pr@c}{#1}%
2828 <debug>\MT@info{1}{creating protrusion list `~\MT@pr@c@name'}%
2829 \def\MT@permute@list{pr@c}%
2830 \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>`, ...

```
2831 \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.

```
2832 \MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}%
2833 \MT@end@catcodes
2834 }
```

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

```
2835 \def\SetExpansion{%
```

```

2836 \MT@begin@catcodes
2837 \MT@SetExpansion
2838 }

\MT@SetExpansion
  \MT@ex@name 2839 \newcommand*\MT@SetExpansion[3] [] {%
\MT@extra@context 2840 \let\MT@extra@context\@empty
2841 \MT@set@named@keys{MT@ex@c}{#1}%
  \MT@permutelist 2842 \MT@ifdefined@n@T{MT@ex@c@MT@ex@c@name @factor}{%
2843 \ifnum\c@name MT@ex@c@MT@ex@c@name @factor\endc@name > \@m
2844 \MT@warning@n{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
2845 too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
2846 maximum of 1000}%
2847 \MT@gl@et@nc{MT@ex@c@MT@ex@c@name @factor}\@m
2848 \fi
2849 }%
2850 <debug>\MT@info{1}{creating expansion list `~\MT@ex@c@name'}%
2851 \def\MT@permutelist{ex@c}%
2852 \setkeys{MT@c@f}{#2}%
2853 \MT@permute
2854 \MT@gdef@n{MT@ex@c@MT@ex@c@name}{#3}%
2855 \MT@end@catcodes
2856 }

\SetTracking
2857 \def\SetTracking{%
2858 \MT@begin@catcodes
2859 \MT@SetTracking
2860 }

\MT@SetTracking Third argument may be empty.
2861 \newcommand*\MT@SetTracking[3] [] {%
2862 \let\MT@extra@context\@empty
2863 \MT@set@named@keys{MT@tr@c}{#1}%
2864 <debug>\MT@info{1}{creating tracking list `~\MT@tr@c@name'}%
2865 \def\MT@permutelist{tr@c}%
2866 \setkeys{MT@c@f}{#2}%
2867 \MT@permute
2868 \KV@sp@def\@tempa{#3}%
2869 \MT@ifempty\@tempa\relax{%
2870 \MT@ifint\@tempa
2871 {\MT@xdef@n{MT@tr@c@MT@tr@c@name}{\@tempa}}%
2872 {\MT@warning{Value `~\@tempa' is not a number in\MessageBreak
2873 tracking set `~\MT@curr@set@name'}}}%
2874 \MT@end@catcodes
2875 }

\SetExtraSpacing
2876 \def\SetExtraSpacing{%
2877 \MT@begin@catcodes
2878 \MT@SetExtraSpacing
2879 }

\MT@SetExtraSpacing
  \MT@sp@c@name 2880 \newcommand*\MT@SetExtraSpacing[3] [] {%
\MT@extra@context 2881 \let\MT@extra@context\@empty
2882 \MT@set@named@keys{MT@sp@c}{#1}%
  \MT@permutelist 2883 <debug>\MT@info{1}{creating spacing list `~\MT@sp@c@name'}%
2884 \def\MT@permutelist{sp@c}%
2885 \setkeys{MT@c@f}{#2}%
2886 \MT@permute
2887 \MT@gdef@n{MT@sp@c@MT@sp@c@name}{#3}%

```

```

2888 \MT@end@catcodes
2889 }

\SetExtraKerning
2890 \def\SetExtraKerning{%
2891 \MT@begin@catcodes
2892 \MT@SetExtraKerning
2893 }

\MT@SetExtraKerning
\MT@kn@c@name 2894 \newcommand*\MT@SetExtraKerning[3] [] {%
\MT@extra@context 2895 \let\MT@extra@context\@empty
2896 \MT@set@named@keys\MT@kn@c\{#1}%
\MT@permutelist 2897 (debug)\MT@edinfo{1}{creating kerning list `\'MT@kn@c@name'%
2898 \def\MT@permutelist\MT@kn@c}%
2899 \setkeys\MT@c@fg\{#2}%
2900 \MT@permute
2901 \MT@gdefn\MT@kn@c@\MT@kn@c@name\{#3}%
2902 \MT@end@catcodes
2903 }

\MT@set@named@keys \MT@options We first set the name (if specified), then remove it from the list, and set the
remaining keys.
2904 \def\MT@set@named@keys#1#2{%
2905 \def\x##1name=##2,##3\@nil{%
2906 \setkeys\{#1\}\{name=##2\}%
2907 \gdef\MT@options\{##1##3\}%
2908 \MT@rem@from@clist\{name=\}\MT@options
2909 }%
2910 \x#2,name=,\@nil
2911 \@expandtwoargs\setkeys\{#1\}\MT@options
2912 }

\MT@define@code@key Define the keys for the configuration lists (which are setting the codes, in pdfTeX
speak).
2913 \def\MT@define@code@key#1#2{%
2914 \define@key\MT@#2\{#1\} [] {%
2915 \@tempcnta=\@ne
2916 \MT@map@clistn\{##1\}%
2917 \KV@sp@def\MT@val\{###1\}%

Here, too, we allow for something like 'bf*'. It will be expanded immediately.
2918 \MT@get@highlevel\{#1\}%
2919 \MT@edefn\MT@temp#1\the\@tempcnta\{\MT@val\}%
2920 \advance\@tempcnta \@ne
2921 }%
2922 }%
2923 }

\MT@define@code@key@size \MT@temp size must be in a \csname, so that it is at least \relax, not undefined.
2924 \def\MT@define@code@key@size#1{%
2925 \define@key\MT@#1\{size\} [] {%
2926 \MT@map@clistn\{##1\}%
2927 \KV@sp@def\MT@val\{###1\}%
2928 \expandafter\MT@get@range\MT@val--\@nil
2929 \ifx\MT@val\relax \else
2930 \MT@exp@cs\MT@xadd\MT@temp size\%
2931 \{\{\MT@lower\}\MT@upper\}\MT@curr@set@name\}\}%
2932 \fi
2933 }%
2934 }%
2935 }

```

\MT@define@code@key@font

```

2936 \def\MT@define@code@key@font#1{%
2937   \define@key{MT@#1}{font}[]{%
2938     \MT@map@clist@{##1}{%
2939       \KV@esp@def\MT@val{###1}%
2940       \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/}}\relax
2941       \expandafter\MT@get@font@and@size\MT@val///// \@nil
2942       \MT@xdef@n{MT@MT@permutelist @\@tempb\MT@extra@context}%
2943       {\cname MT@MT@permutelist @name\endcsname}%
2944       (debug)\MT@info@n{1}{initialising: use list for font \@tempb=\MT@val
2945       (debug)          \ifx\MT@extra@context\@empty\else\MessageBreak
2946       (debug)          (context: \MT@extra@context)\fi}%
2947       \MT@exp@cs\MT@xaddb
2948       {MT@MT@permutelist @\@tempb\MT@extra@context @size}%
2949       {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
2950     }%
2951   }%
2952 }
```

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

```

2953 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2954   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
2955 }

2956 \MT@define@code@key{encoding}{cfg}
2957 \MT@define@code@key{family}   {cfg}
2958 \MT@define@code@key{series}   {cfg}
2959 \MT@define@code@key{shape}     {cfg}
2960 \MT@define@code@key@size      {cfg}
2961 \MT@define@code@key@font      {cfg}
```

\MT@define@opt@key

```

2962 \def\MT@define@opt@key#1#2{%
2963   \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
2964     \MT@xdef@n{MT@#1@c@MT@curr@set@name @#2}{##1}}%
2965 }
```

The options in the optional first argument.

```
2966 \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.

```

2967   \define@key{MT@#1@c}{name}[]{%
2968     \MT@ifempty{##1}{%
2969       \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2970     }{%
2971       \MT@edef@n{MT@#1@c@name}{##1}%
2972       \MT@ifdefined@n@T{MT@#1@c@\cname MT@#1@c@name\endcsname}{%
2973         \MT@warning{Redefining \@nameuse{MT@abbr@#1} list ~\@nameuse{MT@#1@c@name}}%
2974       }%
2975     }%
2976     \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
2977   }%
2978   \MT@define@opt@key{#1}{load}%
2979   \MT@define@opt@key{#1}{factor}%
2980   \MT@define@opt@key{#1}{preset}%
2981   \MT@define@opt@key{#1}{inputenc}%
```

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

```

2982   \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
2983 }
```


Automatically enable font copying if we find a protrusion or expansion context. After the preamble, check whether font copying is enabled. For older pdfTeX versions, disallow. Also disable for luaTeX.

```

2984 \MT@requires@pdftex7{
2985   {lua}
2986   \MT@requires@luatex{
2987     \define@key{MT@ex@c}{context}[]{}%
2988     \MT@error{Expansion contexts currently don't work with luatex.\MessageBreak
2989       Ignoring `context' key\on@line}%
2990     {Use pdftex instead.}%
2991   }
2992 }{
2993 /lua
2994   \define@key{MT@ex@c}{context}[]{}%
2995   \MT@ifempty{#1}\relax{%
2996     \MT@gllet\MT@copy@font\MT@copy@font@
2997     \def\MT@extra@context{#1}%
2998   }%
2999 }
3000 \MT@addto@setup{%
3001   \define@key{MT@ex@c}{context}[]{}%
3002   \ifx\MT@copy@font\MT@copy@font@
3003     \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3004   \else
3005     \MT@error{\MT@MT space isn't set up for expansion contexts.\MessageBreak
3006       Ignoring `context' key\on@line}%
3007     {Either move the settings inside the preamble,\MessageBreak
3008       or load the package with the `copyfonts' option.}%
3009   \fi
3010 }%
3011 }

```

Protrusion contexts *may* also work without copying the font, so we don't issue an error but only a warning. The problem is that pdfTeX only allows one set of protrusion factors for a given font within one paragraph (those that are in effect at the end of the paragraph will be in effect for the whole paragraph). When different fonts are loaded – like in the example with the footnote markers – we don't need to copy the fonts.

```

3012   \define@key{MT@pr@c}{context}[]{}%
3013   \MT@ifempty{#1}\relax{%
3014     \MT@gllet\MT@copy@font\MT@copy@font@
3015     \def\MT@extra@context{#1}%
3016   }%
3017 }
3018 \MT@addto@setup{%
3019   \define@key{MT@pr@c}{context}[]{}%
3020   \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3021   \ifx\MT@copy@font\MT@copy@font@
3022     \MT@warning@nl{If protrusion contexts don't work as expected,
3023       \MessageBreak load the package with the `copyfonts' option}%
3024   \fi
3025 }%
3026 }
3027 {lua} }
3028 }{
3029   \define@key{MT@ex@c}{context}[]{}%
3030   \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
3031     or later. Ignoring `context' key\on@line}%
3032   {Upgrade pdftex.}%
3033 }

```

```
3034 }
```

```
\MT@warn@nodim
```

```
3035 \def\MT@warn@nodim#1{%
3036   \MT@warning{\@tempa' is not a dimension.\MessageBreak
3037             Ignoring it and setting values relative to\MessageBreak #1}%
3038 }
```

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

```
3039 \define@key{MT@pr@c}{unit}[character]{%
3040   \MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
3041   \def\@tempa{#1}%
3042   \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.

```
3043   \MT@ifdimen\@tempa
3044   {\MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
3045   {\MT@warn@nodim{character widths}}%
3046   }%
3047 }
```

Tracking may only be relative to a dimension.

```
3048 \define@key{MT@tr@c}{unit}[1em]{%
3049   \MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
3050   \def\@tempa{#1}%
3051   \MT@ifdimen\@tempa
3052   {\MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
3053   {\MT@warn@nodim{1em}%
3054   \MT@gdefn{MT@tr@c@MT@curr@set@name @unit}{1em}}%
3055 }
```

Spacing and kerning codes may additionally be relative to space dimensions.

```
3056 \MT@map@clist@n{sp,kn}{%
3057   \define@key{MT@#1@c}{unit}[space]{%
3058     \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
3059     \def\@tempa{##1}%
3060     \MT@ifstreq\@tempa{character}\relax{%
3061       \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
3062       \MT@ifstreq\@tempa{space}\relax{%
3063         \MT@ifdimen\@tempa
3064         {\MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
3065         {\MT@warn@nodim{width of space}}%
3066       }%
3067     }%
3068   }%
3069 }
```

The first argument to `\SetExpansion` accepts some more options.

```
3070 \MT@map@clist@n{stretch,shrink,step}{%
3071   \define@key{MT@ex@c}{#1}[]{%
3072     \MT@ifempty{##1}\relax{%
3073       \MT@ifint{##1}{%
```

A space terminates the number.

```
3074       \MT@gdefn{MT@ex@c@MT@curr@set@name @#1}{##1 }%
3075     }%
3076     \MT@warning{%
3077       Value `##1' for option `#1' is not a number.\MessageBreak
3078       Ignoring it}%
3079   }%
3080 }
```

```

3081 }%
3082 }
3083 \define@key{MT@ex@ec}{auto}[true]{%
3084   \def\@tempa{#1}%
3085   \csname if\@tempa\endcsname
3086     \MT@requires@pdftex4{%
3087       \MT@gdefn{MT@ex@ec\MT@curr@set@name @auto}{autoexpand}%
3088     }%
3089     \MT@warning{pdftex too old for automatic font expansion}%
3090   }
3091 \else
3092   \MT@requires@pdftex4{%
3093     \MT@glet@nc{MT@ex@ec\MT@curr@set@name @auto}\@empty
3094   }\relax
3095 \fi
3096 }

```

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

Tracking: Interword spacing and outer kerning. The variant with space in case \SetTracking is called inside an argument (e. g., to \IfFileExists).

```

3097 \MT@define@opt@key{tr}{spacing}
3098 \MT@define@opt@key{tr}{outerspacing}
3099 \MT@define@opt@key{tr}{outerkerning}

```

Which ligatures should be disabled?

```

3100 \define@key{MT@tr@ec}{noligatures}[]%
3101   {\MT@xdefn{MT@tr@ec\MT@curr@set@name @noligatures}{#1}}
3102 \define@key{MT@tr@ec}{outer spacing}[]{\setkeys{MT@tr@ec}{outerspacing={#1}}}
3103 \define@key{MT@tr@ec}{outer kerning}[]{\setkeys{MT@tr@ec}{outerkerning={#1}}}
3104 \define@key{MT@tr@ec}{no ligatures}[]{\setkeys{MT@tr@ec}{noligatures={#1}}}

```

14.3.6 Character inheritance

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

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

```

3105 \renewcommand*\DeclareCharacterInheritance[1][]{%
3106   \let\MT@extra@context\@empty
3107   \let\MT@extra@inputenc\@undefined
3108   \let\MT@inh@feat\@empty
3109   \setkeys{MT@inh@}{#1}%
3110   \MT@begin@catcodes
3111   \MT@set@inh@list
3112 }

```

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

```

3113 \def\MT@set@inh@list#1#2{%
3114   \MT@ifempty\MT@inh@feat{%
3115     \MT@map@clist@c\MT@features{\MT@declare@char@inh{##1}{#1}{#2}}}%
3116   }%
3117   \MT@map@clist@c\MT@inh@feat{%
3118     \KV@esp@def\@tempa{##1}%

```

```

3119 \MT@ifempty\@tempa\relax{%
3120 \MT@exp@one@n\MT@declare@char@inh
3121 {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
3122 }%
3123 }%
3124 }%
3125 \MT@end@catcodes
3126 }

```

The keys for the optional argument.

```

3127 \MT@map@clist@c\MT@features@long{%
3128 \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}%
3129 \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}

```

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

```

3130 \def\MT@declare@char@inh#1#2#3{%
3131 \MT@edef@n{MT@#1@inh@name}%
3132 {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3133 \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3134 \MT@ifdefined@c@T\MT@extra@inputenc{%
3135 \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3136 <debug>\MT@info{1}{creating inheritance list \@nameuse{MT@#1@inh@name}'}%
3137 \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
3138 \def\MT@permutelist{#1@inh}%
3139 \setkeys{MT@inh}{#2}%
3140 \MT@permute
3141 }

```

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

```

3142 \define@key{MT@inh}{encoding}[]{%
3143 \def\MT@val{#1}%
3144 \expandafter\MT@encoding@check\MT@val,\@nil
3145 \MT@get@highlevel{encoding}%
3146 \MT@edef@n{MT@tempencoding1}{\MT@val}%
3147 }

```

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

```

3148 \def\MT@encoding@check#1,#2\@nil{%
3149 \MT@ifempty{#2}\relax%
3150 \edef\MT@val{#1}%
3151 \MT@warning{You may only specify one encoding for character\MessageBreak
3152 inheritance lists. Ignoring encoding(s) #2}%
3153 }%
3154 }

```

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

```

3155 \MT@define@code@key{family}{inh}
3156 \MT@define@code@key{series}{inh}
3157 \MT@define@code@key{shape}{inh}
3158 \MT@define@code@key@size{inh}
3159 \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).

```

3160 \def\MT@inh@do#1,{%
3161 \ifx\relax#1\@empty \else
3162 \MT@inh@split #1==\relax

```

```

3163     \expandafter\MT@inh@do
3164     \fi
3165 }

\MT@inh@split    Only gather the inheriting characters here. Their codes will actually be set in
                  \MT@set@{feature}@codes.
3166 \def\MT@inh@split#1=#2=#3\relax{%
3167     \def\@tempa{#1}%
3168     \ifx\@tempa\@empty \else
3169         \MT@get@slot
3170         \ifnum\MT@char > \m@ne
3171             \let\MT@val\MT@char
3172             \MT@map@clist@n{#2}{%
3173                 \def\@tempa{##1}%
3174                 \ifx\@tempa\@empty \else
3175                     \MT@get@slot
3176                     \ifnum\MT@char > \m@ne
3177                         \MT@exp@cs\MT@xadd\MT@inh@\MT@listname @\MT@val @{{\MT@char}}%
3178                     \fi
3179                 \fi
3180             }%
3181     <debug>\MT@info@n1{2}{children of #1 (\MT@val):
3182     <debug>                                     \@nameuse{\MT@inh@\MT@listname @\MT@val @}}%
3183     \fi
3184     \fi
3185 }

```

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}@/{font axes}@sizes`, which in turn contains the respective *list name*s attached to the ranges.

`\MT@permute@@`

`\MT@permute@@@`

```

3186 \def\MT@permute{%
3187     \let\MT@cnt@encoding\@ne
3188     \MT@permute@

    Undefine commands for the next round.
3189     \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
3190     \MT@gl@t\MT@temp@size\@undefined
3191 }
3192 \def\MT@permute@{%
3193     \let\MT@cnt@family\@ne
3194     \MT@permute@@
3195     \MT@increment\MT@cnt@encoding
3196     \MT@ifdefined@n@T{\MT@temp@encoding\MT@cnt@encoding}%
3197     \MT@permute@
3198 }
3199 \def\MT@permute@@{%
3200     \let\MT@cnt@series\@ne
3201     \MT@permute@@@
3202     \MT@increment\MT@cnt@family
3203     \MT@ifdefined@n@T{\MT@temp@family\MT@cnt@family}%
3204     \MT@permute@@@
3205 }
3206 \def\MT@permute@@@{%
3207     \let\MT@cnt@shape\@ne
3208     \MT@permute@@@
3209     \MT@increment\MT@cnt@series

```

```

3210 \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
3211 \MT@permute@@@
3212 }
3213 \def\MT@permute@@@{%
3214 \MT@permute@@@
3215 \MT@increment\MT@cnt@shape
3216 \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
3217 \MT@permute@@@
3218 }

```

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

```

3219 \def\MT@permute@@@{%
3220 \MT@permute@define{encoding}%
3221 \ifMT@document
3222 \ifx\MT@tempencoding\@empty \else
3223 \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
3224 {\expandafter\expandafter\expandafter\@gobble}%
3225 \fi
3226 \fi
3227 \MT@permute@@@
3228 }

```

\MT@permute@@@

```

3229 \def\MT@permute@@@@{%
3230 \MT@permute@define{family}%
3231 \MT@permute@define{series}%
3232 \MT@permute@define{shape}%
3233 \edef\@tempa{\MT@tempencoding
3234 \MT@tempfamily
3235 \MT@tempseries
3236 \MT@tempshape
3237 \MT@ifdefined@c@T\MT@tempsize *}%

```

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

```

3238 \MT@ifstreq\@tempa{////}\relax%
3239 \ifx\MT@tempencoding\@empty
3240 \MT@warning{%
3241 You have to specify an encoding for\MessageBreak
3242 \@nameuse{MT@abbr@MT@permutelist} list
3243 ~\@nameuse{MT@MT@permutelist @name}'.\MessageBreak
3244 Ignoring it}%
3245 \else
3246 \MT@ifdefined@c@TF\MT@tempsize{%

```

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

```

3247 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3248 \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3249 }%
3250 \MT@exp@cs\MT@xaddb
3251 {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3252 \MT@tempsize
3253 <debug>\MT@info@n1{1}{initialising: use list for font \@tempa,\MessageBreak
3254 <debug> sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3255 <debug> @sizes\endcsname}%
3256 }{%

```

Only one list can apply to a given combination.

```

3257 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3258 \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3259 ~\@nameuse{MT@MT@permutelist @name}' will override list\MessageBreak

```

```

3260         \nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3261         for font \@tempa'%
3262     }%
3263 <debug>\MT@edinfo@n1{1}{initialising: use list for font \@tempa
3264 <debug>         \ifx\MT@extra@context\@empty\else\MessageBreak
3265 <debug>         (context: \MT@extra@context)\fi}%
3266     }%
3267     \MT@xdefn{MT@MT@permutelist @\@tempa\MT@extra@context}%
3268     {\csname MT@MT@permutelist @name\endcsname}%
3269     \fi
3270 }%
3271 }

```

\MT@permute@define Define the commands.

```

3272 \def\MT@permute@define#1{%
3273     \@tempcnta=\csname MT@cnt@#1\endcsname\relax
3274     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3275     {\MT@edefn{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3276     {\MT@let@nc{MT@temp#1}\@empty}%
3277 }

```

\MT@permute@reset Reset the commands.

```

3278 \def\MT@permute@reset#1{%
3279     \@tempcnta=\@ne
3280     \MT@loop
3281     \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3282     \advance\@tempcnta\@ne
3283     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3284     \iftrue
3285     \iffalse
3286     \MT@repeat
3287 }

```

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

```

3288 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

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

```

3289 \def\MT@check@rlist@#1#2#3{%
3290     \def\@tempb{#1}%
3291     \def\@tempc{#2}%
3292     \MT@if@false
3293     \MT@exp@cs\MT@map@tlist@c
3294     {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3295     \MT@check@range
3296 }

```

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

```

3297 \def\MT@check@range#1{\expandafter\MT@check@range@ #1}

```

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

```

3298 \def\MT@check@range@#1#2#3{%
3299     \MT@ifdim{#2}=\m@ne{%
3300         \MT@ifdim\@tempc=\m@ne{%

```

- Both items are simple sizes.

```

3301         \MT@ifdim\@tempb={#1}\MT@if@true\relax
3302     }{%

```

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

```

3303      \MT@ifdim\@tempb>{#1}\relax{%
3304      \MT@ifdim\@tempc>{#1}{%
3305      \MT@if@true
3306      \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3307      }\relax
3308      }%
3309      }%
3310      }{%
3311      \MT@ifdim\@tempc=\m@ne{%

```

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

```

3312      \MT@ifdim\@tempb<{#2}{%
3313      \MT@ifdim\@tempb<{#1}\relax\MT@if@true
3314      }\relax
3315      }{%

```

- Both items are ranges.

```

3316      \MT@ifdim\@tempb<{#2}{%
3317      \MT@ifdim\@tempc>{#1}{%
3318      \MT@if@true
3319      \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3320      }\relax
3321      }\relax
3322      }%
3323      }%
3324      \ifMT@if@
3325      \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3326      \@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
3327      list `#3' for font \@tempa,\MessageBreak size \@tempb}%

```

If we've already found a conflict with this item, we can skip the rest of the list.

```

3328      \expandafter\MT@tlist@break
3329      \fi
3330      }

```

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 3331 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 3332 \newif\ifMT@opt@auto
3333 \newif\ifMT@opt@DVI

\MT@optwarn@admissible    Some warnings.
3334 \def\MT@optwarn@admissible#1#2{%
3335   \MT@warning@nl{`#1' is not an admissible value for option\MessageBreak
3336   `#2'. Assuming `false'}%
3337 }

\MT@optwarn@nan
3338 </package>
3339 <plain>\MT@requires@latex1{
3340 \def\MT@optwarn@nan#1#2{%
3341   \MT@warning@nl{Value `#1' for option `#2' is not a\MessageBreak number.
3342   Using default value of \number\@nameuse{MT@#2@default}}%

```



```

3343 }
3344 <plain>\relax
3345 <*package>

```

\MT@opt@def@set

```

3346 \def\MT@opt@def@set#1{%
3347   \MT@ifdefined@n@TF{MT@ \@tempb @set@{MT@val}}{%
3348     \MT@xdef@n{MT@ \@tempb @setname}{\MT@val}%
3349   }{%
3350     \MT@xdef@n{MT@ \@tempb @setname}{\@nameuse{MT@default@ \@tempb @set}}%
3351     \MT@warning@n{The #1 set `MT@val' is undeclared.\MessageBreak
3352                   Using set ` \@nameuse{MT@ \@tempb @setname}' instead}%
3353   }%
3354 }

```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a <set name>.

```

3355 \MT@map@clist@n{protrusion,expansion}{%
3356   \define@key{MT}{#1}[true]{%
3357     \csname MT@opt@#1true\endcsname
3358     \MT@map@clist@n{##1}{%
3359       \KV@sp@def\MT@val{###1}%
3360       \MT@ifempty\MT@val\relax{%
3361         \csname MT@#1true\endcsname
3362         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3363         \MT@ifstreq\MT@val{true}\relax
3364       }%
3365       \MT@ifstreq\MT@val{false}{%
3366         \csname MT@#1false\endcsname
3367       }{%
3368         \MT@ifstreq\MT@val{compatibility}{%
3369           \MT@let@nc{MT@ \@tempb @level}\@ne
3370         }{%
3371           \MT@ifstreq\MT@val{nocompatibility}{%
3372             \MT@let@nc{MT@ \@tempb @level}\tw@
3373           }%

```

If everything failed, it should be a set name.

```

3374       \MT@opt@def@set{#1}%
3375     }%
3376   }%
3377 }%
3378 }%
3379 }%
3380 }%
3381 }%
3382 }

```

activate is a shortcut for protrusion and expansion.

```

3383 \define@key{MT}{activate}[true]{%
3384   \setkeys{MT}{protrusion={#1}}%
3385   \setkeys{MT}{expansion={#1}}%
3386 }

```

spacing, kerning and tracking do not have a compatibility level.

```

3387 \MT@map@clist@n{spacing,kerning,tracking}{%
3388   \define@key{MT}{#1}[true]{%
3389     \MT@map@clist@n{##1}{%
3390       \KV@sp@def\MT@val{###1}%
3391       \MT@ifempty\MT@val\relax{%
3392         \csname MT@#1true\endcsname
3393       \MT@ifstreq\MT@val{true}\relax

```

```

3394     {%
3395     \MT@ifstreq\MT@val{false}{%
3396     \csname MT@#1false\endcsname
3397     }{%
3398     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3399     \MT@opt@def@set{#1}%
3400     }%
3401     }%
3402     }%
3403     }%
3404     }%
3405 }

```

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

```

3406 \def\MT@def@bool@opt#1#2{%
3407   \define@key{MT}{#1}[true]{%
3408     \def\@tempa{##1}%
3409     \MT@ifstreq\@tempa{true}\relax{%
3410     \MT@ifstreq\@tempa{false}\relax{%
3411     \MT@optwarn@admissible{##1}{#1}%
3412     \def\@tempa{false}%
3413     }%
3414     }%
3415     #2%
3416     }%
3417 }

```

Boolean options that only set the switch.

```

3418 \MT@map@clist@n{draft,selected,babel}{%
3419   \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
3420 \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.

```

3421 \MT@def@bool@opt{DVIoutput}{%
3422   \csname if\@tempa\endcsname
3423   \ifnum\pdfoutput>\z@ \MT@opt@DVIttrue \fi
3424   \pdfoutput\z@
3425   \else
3426   \ifnum\pdfoutput<\@ne \MT@opt@DVIttrue \fi
3427   \pdfoutput\@ne
3428   \fi
3429 }

```

Setting the deferssetup 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.

```

3430 \MT@def@bool@opt{deferssetup}{%
3431   \csname if\@tempa\endcsname \else
3432   \AtEndOfPackage{%
3433     \MT@setup@
3434     \let\MT@setup@\@empty
3435     \let\MT@addto@setup\@firstofone
3436     }%
3437   \fi
3438 }

```

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.

```

3439 \MT@requires@pdftex7{
3440   (*lua)
3441   \MT@requires@luatex{
3442     \MT@def@bool@opt{copyfonts}{%
3443       \csname if\@tempa\endcsname
3444       \MT@error{The `copyfonts' option doesn't work with luatex}
3445       {Use pdftex instead.}%
3446     }
3447   }
3448 }{
3449   (/lua)
3450   \MT@def@bool@opt{copyfonts}{%
3451     \csname if\@tempa\endcsname
3452     \MT@gllet\MT@copy@font\MT@copy@font@
3453     \else
3454     \MT@gllet\MT@copy@font\relax
3455     \fi
3456   }
3457 }{lua} }
3458 }{
3459   \MT@def@bool@opt{copyfonts}{%
3460     \csname if\@tempa\endcsname
3461     \MT@error{The pdftex version you are using is too old\MessageBreak
3462       to use the `copyfonts' option}{Upgrade pdftex.}%
3463     \fi
3464   }
3465 }

```

final is the opposite to draft.

```

3466 \MT@def@bool@opt{final}{%
3467   \csname if\@tempa\endcsname
3468   \MT@draftfalse
3469   \else
3470   \MT@drafttrue
3471   \fi
3472 }

```

For verbose output, we redefine \MT@vinfo.

```

3473 \define@key{MT}{verbose}[true]{%
3474   \let\MT@vinfo\MT@info@n1
3475   \def\@tempa{#1}%
3476   \MT@ifstreq\@tempa{true}\relax{%

```

Take problems seriously.

```

3477   \MT@ifstreq\@tempa{errors}{%
3478     \let\MT@warning \MT@warn@err
3479     \let\MT@warning@n1\MT@warn@err
3480   }{%
3481     \let\MT@vinfo\@gobble

```

Cast warnings to the winds.

```

3482   \MT@ifstreq\@tempa{silent}{%
3483     \let\MT@warning \MT@info
3484     \let\MT@warning@n1\MT@info@n1
3485   }{%
3486     \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
3487   }%
3488 }%
3489 }%

```

3490 }

Options with numerical keys: factor, stretch, shrink, step, letterspace.

```
3491 </package>
3492 <plain>\MT@requires@latex1{
3493 \MT@map@clist@n{%
3494 <package> stretch,shrink,step,%
3495 letterspace}{%
3496 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
3497 \def\@tempa{##1 }%
```

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

```
3498 \MT@ifint\@tempa
3499 {\MT@edef@n{MT@#1}{\@tempa}}%
3500 {\MT@optwarn@nan{##1}{#1}}%
3501 }%
3502 }
3503 <plain>\relax
3504 <*package>
```

factor will define the protrusion factor only.

```
3505 \define@key{MT}{factor}[\MT@factor@default]{%
3506 \def\@tempa{#1 }%
3507 \MT@ifint\@tempa
3508 {\edef\MT@pr@factor{\@tempa}}
3509 {\MT@optwarn@nan{#1}{factor}}%
3510 }
```

Unit for protrusion codes.

```
3511 \define@key{MT}{unit}[character]{%
3512 \def\@tempa{#1}%
3513 \MT@ifstreql\@tempa{character}\relax{%
3514 \MT@ifdimen\@tempa
3515 {\let\MT@pr@unit\@tempa}%
3516 {\MT@warning@n1{\@tempa' is not a dimension.\MessageBreak
3517 Ignoring it and setting values relative to\MessageBreak
3518 character widths}}%
3519 }%
3520 }
```

14.4.2 Reading the configuration file

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as 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 .)

```
3521 \MT@protrusiontrue
3522 \ifnum\pdfoutput<\@ne \else
```

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

```
3523 \MT@requires@pdftex4{
3524 \MT@expansiontrue
3525 \MT@autottrue
3526 }\relax
3527 \fi
```

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

`\MT@config@file` However, the config option must of course be evaluated beforehand. We also have
`\MT@get@config` to define a no-op for the regular option processing later.

```

3528 \define@key{MT}{config}[]{\relax}
3529 \def\MT@get@config#1config=#2,#3\@nil{%
3530   \MT@ifempty{#2}%
3531   {\def\MT@config@file{\MT@MT.cfg}}%
3532   {\def\MT@config@file{#2.cfg}}%
3533 }
3534 \expandafter\expandafter\expandafter\MT@get@config
3535 \csname opt@\@currname.\@current\endcsname,config=,\@nil

```

Load the file.

```

3536 \IfFileExists{\MT@config@file}{%
3537   \MT@info@nl{Loading configuration file \MT@config@file}%
3538   \MT@begin@catcodes
3539   \let\MT@begin@catcodes\relax
3540   \let\MT@end@catcodes\relax
3541   \let\MT@curr@file\MT@config@file
3542   \input{\MT@config@file}%
3543   \endgroup
3544 }{\MT@warning@nl{%
3545   Could not find configuration file '\MT@config@file'!\MessageBreak
3546   This will almost certainly cause undesired results.\MessageBreak
3547   Please fix your installation}%
3548 }

```

`\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).

```

3549 \def\MT@check@active@set#1{%
3550   \MT@ifdefined@n@TF{MT@#1@setname}{%
3551     \MT@info@nl{Using \@nameuse{MT@abbr@#1} set '\@nameuse{MT@#1@setname}'}%
3552   }{%
3553     \MT@ifdefined@n@TF{MT@default@#1@set}{%
3554       \MT@glet@nn{MT@#1@setname}{MT@default@#1@set}%
3555       \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set '\@nameuse{MT@#1@setname}'}%
3556     }{%

```

If no default font set has been declared in the main configuration file, we use the
(empty, non-existent) set '0', and issue a warning.

```

3557   \MT@gdef@n{MT@#1@setname}{0}%
3558   \MT@warning@nl{No \@nameuse{MT@abbr@#1} set chosen, no default set declared.
3559     \MessageBreak Using empty set}%
3560   }%
3561 }%
3562 }

```

14.4.3 Hook for other packages

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

This hook was needed in versions prior to 1.9a to overcome the situation that
(1) the microtype package should be loaded after all font defaults have been set
up (hence, using `\ifpackageloaded` in the font package was not viable), and
(2) checking `\AtBeginDocument` could be too late, since fonts might already have
been loaded, and consequently set up, in the preamble. With the new deferred

setup, one could live without this command, however, it remains here since it's simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

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

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

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

```
3563 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
3564   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3565   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
3566 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook
```

14.4.4 Changing options later

`\microtypesetup` Inside the preamble, `\microtypesetup` accepts the same options as the package (unless `defersetup=false`). In the document body, it accepts the options: `protrusion`, `expansion`, `activate`, `tracking`, `spacing` and `kerning`. Specifying font sets is not allowed.

```
3567 \def\microtypesetup{\setkeys{MT}}
3568 \MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
3569 \def\MT@define@optionX#1#2{%
3570   \define@key{MTX}{#1}[true]{%
3571     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3572     \MT@map@clist@n{##1}{%
3573       \KV@@sp@def\MT@val{###1}%
3574       \MT@ifempty\MT@val\relax{%
3575         \@tempcnta=\m@ne
3576         \MT@ifstreq\MT@val{true}{%

```

Enabling micro-typography in the middle of the document is not allowed if it has been disabled in the package options since fonts might already have been loaded and hence wouldn't be set up.

```
3577   \MT@checksetup{#1}{%
3578     \@tempcnta=\csname MT@\@tempb @level\endcsname
3579     \MT@vinfo{Enabling #1
3580       (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
3581     }%
3582   }{%
3583     \MT@ifstreq\MT@val{false}{%
3584       \@tempcnta=\z@
3585       \MT@vinfo{Disabling #1\on@line}%
3586     }{%
3587       \MT@ifstreq\MT@val{compatibility}{%
3588         \MT@checksetup{#1}{%
3589           \@tempcnta=\@ne
3590           \MT@let@nc{MT@\@tempb @level}\@ne
3591           \MT@vinfo{Setting #1 to level 1\on@line}%
3592         }%
3593       }{%

```

```

3594         \MT@ifstreq\MT@val{nocompatibility}{%
3595             \MT@checksetup{#1}{%
3596                 \@tempcnta=\tw@
3597                 \MT@let@nc{MT@\@tempb @level}\tw@
3598                 \MT@vinfo{Setting #1 to level 2\on@line}%
3599             }%
3600             {%\MT@error{Value `~\MT@val' for key `~#1' not recognised}
3601                 {Use any of `true', `false', `compatibility' or
3602                 `nocompatibility'.}%
3603             }%
3604         }%
3605     }%
3606 }%
3607 \ifnum\@tempcnta>\m@ne
3608     #2\@tempcnta\relax
3609 \fi
3610 }%
3611 }%
3612 }%
3613 }

```

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

```

3614 \def\MT@checksetup#1{%
3615     \csname ifMT@#1\endcsname
3616     \expandafter\@firstofone
3617     \else
3618         \MT@error{You cannot enable #1 if it was disabled\MessageBreak
3619             in the package options}{Load microtype with #1 enabled.}%
3620         \expandafter\@gobble
3621     \fi
3622 }

3623 \MT@define@optionX{protrusion}\pdfprotrudechars
3624 \MT@define@optionX{expansion}\pdfadjustspacing

```

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

```

3625 \MT@requires@pdftex6{
3626     (lua) \MT@requires@luatex\@firstofone{
3627         \def\MT@define@optionX@#1#2{%
3628             \define@key{MTX}{#1}[true]{%
3629                 \MT@map@clist@n{##1}{%
3630                     \KV@sp@def\MT@val{####1}%
3631                     \MT@ifempty\MT@val\relax{%
3632                         \@tempcnta=\m@ne
3633                         \MT@ifstreq\MT@val{true}{%
3634                             \MT@checksetup{#1}{%
3635                                 \@tempcnta=\@ne
3636                                 \MT@vinfo{Enabling #1\on@line}%
3637                             }%
3638                         }{%
3639                             \MT@ifstreq\MT@val{false}{%
3640                                 \@tempcnta=\z@
3641                                 \MT@vinfo{Disabling #1\on@line}%
3642                             }{%\MT@error{Value `~\MT@val' for key `~#1' not recognised}
3643                                 {Use either `true' or `false'.}%
3644                             }%
3645                         }%
3646                     \ifnum\@tempcnta>\m@ne
3647                         #2\relax
3648                     \fi
3649                 }%
3650             }%

```

```

3651 }%
3652 }

```

We cannot simply let `\MT@tracking` relax, since this may select the already letter-spaced font instance.

```

3653 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
3654                               \else \let\MT@tracking\MT@tracking@ \fi}
3655 \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
3656 \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3657                               \pdfappendkern \@tempcnta}
3658 \gobble
3659 \lua{}
3660 }\@firstofone

```

Disable for older pdfTeX versions and for luaTeX.

```

3661 {\define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
3662 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
3663 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
3664 }
3665 \define@key{MTX}{activate}[true]{%
3666   \setkeys{MTX}{protrusion={#1}}%
3667   \setkeys{MTX}{expansion={#1}}%
3668 }

```

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

```

3669 \let\MT@saved@setupfont\MT@setupfont
3670 \define@key{MTX}{disable}[]{}%
3671 \MT@info{Inactivate '\MT@MT' package}%
3672 \let\MT@setupfont\relax
3673 }
3674 \define@key{MTX}{enable}[]{}%
3675 \MT@info{Reactivate '\MT@MT' package}%
3676 \let\MT@setupfont\MT@saved@setupfont
3677 }
3678 \package

```

14.4.5 Processing the options

`\MT@ProcessOptionsWithKV` Parse options.

```

3679 \plain\MT@requires@latex1{
3680 \def\MT@ProcessOptionsWithKV#1{%
3681   \let\@tempc\relax
3682   \let\MT@temp\@empty
3683 \plain\MT@requires@latex2{
3684   \MT@map@clist@c\@classoptionslist{%
3685     \def\CurrentOption{##1}%
3686     \MT@ifdefined@n@T{KV@#1@}{\expandafter\MT@getkey\CurrentOption=\@nil}{%
3687       \edef\MT@temp{\MT@temp,\CurrentOption,}%
3688       \@expandtwoargs\@removeelement\CurrentOption
3689       \@unusedoptionlist\@unusedoptionlist
3690     }%
3691   }%
3692   \edef\MT@temp{\noexpand\setkeys{#1}%
3693     {\MT@temp\@optionlist{\@currname.\@current}}}%

```

`eplain` can handle package options.

```

3694 \plain
3695 {\edef\MT@temp{\noexpand\setkeys{#1}%
3696   {\csname usepkg@options@usepkg@pkg\endcsname}}}

```



```

3697 </plain>
3698 \MT@temp
3699 \MT@clear@options
3700 }

```

\MT@getkey For key=val in class options.

```

3701 \def\MT@getkey#1=#2\@nil{#1}
3702 \MT@ProcessOptionsWithKV{MT}
3703 <plain>\relax
3704 <*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).

```

3705 \MT@addto@setup{%
3706 \ifMT@draft

```

We disable most of what we've just defined in the 3706 lines above if we are running in draft mode.

```

3707 \MT@warning@nl{'draft' option active.\MessageBreak
3708             Disabling all micro-typographic extensions.\MessageBreak
3709             This might lead to different line and page breaks}%
3710 \let\MT@setupfont\relax
3711 \renewcommand*\LoadMicrotypeFile[1]{}%
3712 \renewcommand*\microtypesetup[1]{}%
3713 \renewcommand*\microtypecontext[1]{}%
3714 \renewcommand*\lsstyle{}%
3715 \else

```

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

```

3716 \ifnum\pdfoutput<\@ne
3717 \ifMT@opt@expansion \else
3718 \MT@expansionfalse
3719 \fi
3720 \fi

```

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

Some packages depend on the value of \pdfoutput and will get confused if it is changed after they have been loaded. These packages are, among others: color, graphics, hyperref, crop, contour, pstricks and, as a matter of course, ifpdf. Instead of testing for each package (that's not our job), we only say that it was microtype that changed it. This must be sufficient!

```

3721 \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3722             \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%

```

Working on font copies?

```

3723 \ifx\MT@copy@font\relax\else \MT@info@nl{Using font copies for contexts}\fi

```

Fix the font sets.

```

3724 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set

```

Protrusion.

```

3725 \ifMT@protrusion
3726 \edef\MT@active@features{\MT@active@features,pr}%
3727 \pdfprotrudechars\MT@pr@level
3728 \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)}%

```

```

3729     \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3730         factor: \number\MT@pr@factor\fi
3731     \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
3732     \MT@check@active@set{pr}%
3733 \else
3734     \let\MT@protrusion\relax
3735     \MT@info@nl{No character protrusion}%
3736 \fi

```

Expansion.

```

3737 \ifMT@expansion

```

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

```

3738     \ifnum\MT@stretch=\m@ne
3739         \let\MT@stretch\MT@stretch@default
3740     \fi

```

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

```

3741     \ifnum\MT@shrink=\m@ne
3742         \let\MT@shrink\MT@stretch
3743     \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.

```

3744     \MT@requires@pdftex6{\def\MT@step{1 }}}%
3745     \ifnum\MT@step=\m@ne
3746         \ifnum\MT@stretch>\MT@shrink
3747             \ifnum\MT@shrink=\z@
3748                 \@tempcnta=\MT@stretch
3749             \else
3750                 \@tempcnta=\MT@shrink
3751             \fi
3752         \else
3753             \ifnum\MT@stretch=\z@
3754                 \@tempcnta=\MT@shrink
3755             \else
3756                 \@tempcnta=\MT@stretch
3757             \fi
3758         \fi
3759         \divide\@tempcnta 5\relax
3760     \else
3761         \@tempcnta=\MT@step
3762         \ifnum\@tempcnta=\z@
3763             \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
3764                 Setting it to one}
3765         \fi
3766     \fi
3767     \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3768     \edef\MT@step{\number\@tempcnta\space}}%

```

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

```

3769     \let\MT@auto\@empty
3770     \ifMT@auto
3771         \MT@requires@pdftex4{%

```

We turn off automatic expansion if output mode is DVI.

```

3772 \ifnum\pdfoutput<\@ne
3773 \ifMT@opt@auto
3774 \MT@error{%
3775 Automatic font expansion only works for PDF output.\MessageBreak
3776 However, you are creating a DVI file}
3777 {If you have created expanded fonts instances, remove `auto' from%
3778 \MessageBreak the package options. Otherwise, you have to switch
3779 off expansion\MessageBreak completely.}%
3780 \fi
3781 \MT@autofalse
3782 \else
3783 \def\MT@auto{autoexpand}%
3784 \fi

```

Also, if pdf_TEX is too old.

```

3785 }{%
3786 \MT@error{%
3787 The pdftex version you are using is too old for\MessageBreak
3788 automatic font expansion}%
3789 {If you have created expanded fonts instances, remove `auto' from\MessageBreak
3790 the package options. Otherwise, you have to switch off expansion\MessageBreak
3791 completely, or upgrade pdftex to version 1.20 or newer.}%
3792 \MT@autofalse
3793 \def\MT@auto{1000 }%
3794 }%
3795 \else

```

No automatic expansion.

```

3796 \MT@requires@pdftex4\relax{%
3797 \def\MT@auto{1000 }%
3798 }%
3799 \fi

```

Choose the appropriate macro for selected expansion.

```

3800 \ifMT@selected
3801 \let\MT@set@ex@codes\MT@set@ex@codes@s
3802 \else
3803 \let\MT@set@ex@codes\MT@set@ex@codes@n
3804 \fi

```

Filter out stretch=0, shrink=0, since it would result in a pdf_TEX error.

```

3805 \ifnum\MT@stretch=\z@
3806 \ifnum\MT@shrink=\z@
3807 \MT@warning@n1{%
3808 Both the stretch and shrink limit are set to zero.\MessageBreak
3809 Disabling font expansion}%
3810 \MT@expansionfalse
3811 \fi
3812 \fi
3813 \fi
3814 \ifMT@expansion
3815 \edef\MT@active@features{\MT@active@features,ex}%
3816 \pdfadjustspacing\MT@ex@level
3817 \MT@info@n1{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
3818 (level \number\MT@ex@level),\MessageBreak
3819 stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3820 step: \number\MT@step, \ifMT@selected\else non-\fi selected}%

```

\MT@check@step Check whether stretch and shrink are multiples of step.

```

3821 \def\MT@check@step#1{%
3822 \@tempcnta=\csname MT@#1\endcsname

```

```

3823 \divide\@tempcnta \MT@step
3824 \multiply\@tempcnta \MT@step
3825 \ifnum\@tempcnta=\csname MT@#1\endcsname\else
3826 \MT@warning@nl{The #1 amount is not a multiple of step.\MessageBreak
3827 The effective maximum #1 is \the\@tempcnta\space
3828 (step \number\MT@step)}%
3829 \fi
3830 }%
3831 \MT@check@step{stretch}%
3832 \MT@check@step{shrink}%
3833 \MT@check@active@set{ex}%

```

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

```

3834 \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3835 \color@begingroup\everypar{}\parfillskip\z@skip
3836 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3837 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}%

```

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

```

3838 \gdef\showhyphens#1{\setbox0\vbox{%
3839 \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3840 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3841 \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}%

3842 \else
3843 \let\MT@expansion\relax
3844 \MT@info@nl{No font expansion}%
3845 \fi
3846 }
3847 \MT@requires@pdftex6{

```

`\MT@warn@lua` Switch off the features that don't work with lua_{TeX}.

```

3848 <lua>
3849 \def\MT@warn@lua#1{%
3850 \MT@error{The '~#1' feature doesn't currently work\MessageBreak with luatex}
3851 {Use pdftex instead.}%
3852 \csname MT@#1false\endcsname
3853 \MT@let@nc{MT@#1}\relax
3854 }
3855 </lua>

3856 </package>
3857 \MT@addto@setup{%
3858 <package>

```

Tracking, spacing and kerning.

```

3859 \ifMT@tracking
3860 <lua> \MT@requires@luatex{\MT@warn@lua{tracking}}{%
3861 \edef\MT@active@features{\MT@active@features,tr}%
3862 \MT@info@nl{Tracking enabled}%
3863 \MT@check@active@set{tr}%

```

Enable protrusion for compensation at the line edges.

```

3864 \ifMT@protrusion\else\pdfprotrudechars\@ne\fi
3865 <lua> }%
3866 \else
3867 \let\MT@tracking\relax
3868 \MT@info@nl{No tracking}%
3869 \fi
3870 \ifMT@spacing
3871 <lua> \MT@requires@luatex{\MT@warn@lua{spacing}}{%
3872 \edef\MT@active@features{\MT@active@features,sp}%
3873 \pdfadjustinterwordglue\@ne

```

```

3874      \MT@info@nl{Adjustment of interword spacing enabled}%
3875      \MT@check@active@set{sp}%
3876 \lua      }%
3877      \else
3878      \let\MT@spacing\relax
3879      \MT@info@nl{No adjustment of interword spacing}%
3880      \fi
3881      \ifMT@kerning
3882 \lua      \MT@requires@luatex{\MT@warn@lua{kerning}}}%
3883      \edef\MT@active@features{\MT@active@features,kn}%
3884      \pdfprependkern\@ne
3885      \pdfappendkern\@ne
3886      \MT@info@nl{Adjustment of character kerning enabled}%
3887      \MT@check@active@set{kn}%
3888 \lua      }%
3889      \else
3890      \let\MT@kerning\relax
3891      \MT@info@nl{No adjustment of character kerning}%
3892      \fi
3893 \package

```

\MT@warn@tracking@DVI We issue a warning, when letterspacing in DVI mode, since it will probably not work. We also switch on protrusion if it isn't already, to compensate for the letterspacing kerns.

```

3894      \ifnum\pdfoutput<\@ne
3895      \def\MT@warn@tracking@DVI{%
3896      \MT@warning@nl{%
3897      You are using tracking/letterspacing in DVI mode.\MessageBreak
3898      This will probably not work, unless the post-\MessageBreak
3899      processing program (dvips, dvipdfm(x), ...) is\MessageBreak
3900      able to create the virtual fonts on the fly}%
3901      \MT@glet\MT@warn@tracking@DVI\relax
3902      }%
3903      \else
3904      \def\MT@warn@tracking@DVI{%
3905      \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
3906      \MT@glet\MT@warn@tracking@DVI\relax
3907      }%
3908      \fi
3909      \ifnum\MT@letterspace=\m@ne
3910      \let\MT@letterspace\MT@letterspace@default
3911      \else
3912      \MT@ls@too@large\MT@letterspace
3913      \fi
3914      }%

```

If pdfTeX is too old, we disable tracking, spacing and kerning, and throw an error message.

```

3915 \package
3916 }{
3917 \MT@addto@setup{%
3918 \ifMT@tracking
3919 \MT@error{Tracking only works with pdftex version 1.40\MessageBreak
3920 or newer. Switching it off}{Upgrade pdftex.}%
3921 \else
3922 \MT@info@nl{No tracking (pdftex too old)}%
3923 \fi
3924 \ifMT@spacing
3925 \MT@error{Adjustment of interword spacing only works with\MessageBreak
3926 pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
3927 \else

```

```

3928 \MT@info@n{No adjustment of interword spacing (pdfTeX too old)}%
3929 \fi
3930 \ifMT@kerning
3931 \MT@error{Character kerning only works with\MessageBreak
3932 pdfTeX version 1.40 or newer. Switching it off}{Upgrade pdfTeX.}%
3933 \else
3934 \MT@info@n{No adjustment of character kerning (pdfTeX too old)}%
3935 \fi
3936 }
3937 }

```

`\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```

3938 \MT@requires@pdfTeX5{
3939 \MT@addto@setup{%
3940 \ifMT@noligatures \else
3941 \let\MT@noligatures\relax
3942 \fi
3943 }
3944 }\relax

```

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

```

3945 \MT@addto@setup{%
3946 \ifx\MT@active@features\empty \else
3947 \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
3948 \fi
3949 \MT@documenttrue
3950 }

```

`\MT@set@babel@context` Interaction with babel.

```

3951 \def\MT@set@babel@context#1{%
3952 \MT@ifdefined@n@TF{MT@babel@#1}{%
3953 \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
3954 \expandafter\MT@exp@one@n\expandafter\microtypecontext
3955 \csname MT@babel@#1\endcsname
3956 }{%
3957 \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3958 }%
3959 }

```

`\MT@shorthandoff` Active characters can only be switched off if babel isn't loaded after microtype.

```

3960 \ifpackage@loaded{babel}{
3961 \def\MT@shorthandoff#1#2{%
3962 \MT@info@n{Switching off #1 babel's active characters (#2)}%
3963 \shorthandoff{#2}}
3964 }{
3965 \def\MT@shorthandoff#1#2{%
3966 \MT@error{You must load `babel' before `~\MT@MT'}
3967 {Otherwise, `~\MT@MT' cannot switch off #1 babel's\MessageBreak
3968 active characters.}}
3969 }

```

We patch the language switching commands to enable language-dependent setup.

```

3970 \MT@addto@setup{%
3971 \ifMT@babel
3972 \ifpackage@loaded{babel}{%
3973 \MT@info@n{Redefining babel's language switching commands}%
3974 \let\MT@orig@select@language\select@language
3975 \def\select@language#1{%
3976 \MT@orig@select@language{#1}%
3977 \MT@set@babel@context{#1}%

```

```

3978 }%
3979 \let\MT@orig@foreign@language\foreign@language
3980 \def\foreign@language#1{%
3981   \MT@orig@foreign@language{#1}%
3982   \MT@set@babel@context{#1}%
3983 }%
3984 \ifMT@kerning

```

Disable French babel's active characters.

```

3985 \MT@if@false
3986 \MT@with@babel@and@T{french} \MT@if@true
3987 \MT@with@babel@and@T{frenchb} \MT@if@true
3988 \MT@with@babel@and@T{francais} \MT@if@true
3989 \MT@with@babel@and@T{canadien} \MT@if@true
3990 \MT@with@babel@and@T{acadian} \MT@if@true
3991 \ifMT@if@MT@shorthandoff{French}{:;!}\fi

```

Disable Turkish babel's active characters.

```

3992 \MT@if@false
3993 \MT@with@babel@and@T{turkish} \MT@if@true
3994 \ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi
3995 \fi

```

In case babel was loaded before microtype:

```

3996 \MT@set@babel@context\language
3997 }{%
3998 \MT@warning@nl{You did not load the babel package.\MessageBreak
3999   The `babel' option won't have any effect}%
4000 }%
4001 \fi
4002 }

```

Now we close the \fi from \ifMT@draft.

```

4003 \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.

```

4004 \selectfont}

```

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

```

4005 \edef\MT@curr@file{\jobname.tex}

```

Finally, execute the setup macro at the end of the preamble, and empty it (the combine class calls it repeatedly).

```

4006 /package
4007 plain\MT@requires@latex1{
4008 \AtBeginDocument{\MT@setup@ \MT@gl@et\MT@setup@\@empty}
4009 plain}\relax

```

Warning if \nonfrenchspacing is active, since space factors will be ignored with \pdfadjustinterwordglue>0. Why 1500? Because some packages redefine \frenchspacing.¹⁴ This has to be checked after the setup has taken place. There still will be a false warning if babel is loaded after microtype (without the babel option).

```

4010 *package
4011 \MT@requires@pdftex6{
4012 \AtBeginDocument{%
4013   \ifMT@spacing

```

14 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

```

4014 \ifMT@babel \else
4015 \ifnum\sfcode`. > 1500
4016 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
4017 \MT@warning@n1{%
4018 \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
4019 interword spacing will disable it. You might want\MessageBreak
4020 to add \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
4021 to your preamble}%
4022 }%
4023 \fi
4024 \fi
4025 \fi
4026 }
4027 }\relax
4028 </package>

Restore catcodes.
4029 \MT@restore@catcodes

That was that.
4030 </package>|letterspace>

```

15 Configuration files

Let's now write the font configuration files.

```

4031 <*config>
4032

```

15.1 Font sets

We first declare some sets in the main configuration file.

```

4033 <*m-t>
4034 %%% -----
4035 %%% FONT SETS
4036
4037 \DeclareMicrotypeSet{all}
4038 { }
4039
4040 \DeclareMicrotypeSet{allmath}
4041 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
4042
4043 \DeclareMicrotypeSet{alltext}
4044 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
4045
4046 \DeclareMicrotypeSet{basicmath}
4047 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,OML,OMS},
4048   family   = {rm*,sf*},
4049   series    = {md*},
4050   size      = {normalsize,footnotesize,small,large}
4051 }
4052
4053 \DeclareMicrotypeSet{basictext}
4054 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5},
4055   family   = {rm*,sf*},
4056   series    = {md*},
4057   size      = {normalsize,footnotesize,small,large}
4058 }

```



```

4059

4060 \DeclareMicrotypeSet{smallcaps}
4061   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1},
4062     shape     = {sc*}
4063   }
4064

4065 \DeclareMicrotypeSet{footnotesize}
4066   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1},
4067     size      = {-small}
4068   }
4069
4070 \DeclareMicrotypeSet{scriptsize}
4071   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1},
4072     size      = {-footnotesize}
4073   }
4074
4075 \DeclareMicrotypeSet{normalfont}
4076   { font = */*/*/*/ }
4077

```

The default sets.

```

4078 %%% -----
4079 %%% DEFAULT SETS
4080
4081 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4082 \DeclareMicrotypeSetDefault[expansion]{basictext}
4083 \DeclareMicrotypeSetDefault[spacing]{basictext}
4084 \DeclareMicrotypeSetDefault[kerning]{alltext}
4085 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
4086

```

15.2 Font variants and aliases

```

4087 %%% -----
4088 %%% FONT VARIANTS AND ALIASES
4089

```

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals):

```

4090 \DeclareMicrotypeVariants{x,j,w,a,d,0,1}
4091

```

Other candidates: 2 (proportional digits), e (engraved), f (Fraktur), g (small text), h (shadow), l (outline), n (informal), p (ornaments), r (roman), s (sans serif), t (typewriter). I've omitted them since they seem hardly be used and/or they are actually more than a variant, i. e., they shouldn't share a file.

Fonts that are 'the same': The Latin Modern fonts, the virtual fonts from the ae and zefonts, and the eco and hfoldsty packages (oldstyle numerals) all inherit the (basic) settings from Computer Modern Roman. Some of them are in part overwritten later.

```

4092 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
4093 \DeclareMicrotypeAlias{aer}{cmr} % ae
4094 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
4095 \DeclareMicrotypeAlias{cmor}{cmr} % eco
4096 \DeclareMicrotypeAlias{hfor}{cmr} % hfoldsty

```

The packages `pxfonts` and `txfonts` inherit Palatino and Times settings respectively, also the TeX Gyre fonts Pagella and Termes (formerly: `qfonts`).

```
4097 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
4098 \DeclareMicrotypeAlias{qpl}{ppl} % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)
```

The ‘FPL Neu’ fonts, a ‘re-implementation’ of Palatino.

```
4099 \DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
4100 \DeclareMicrotypeAlias{fp9j}{pplj} % "
4101 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
4102 \DeclareMicrotypeAlias{qtm}{ptm} % TeX Gyre Termes (formerly: qfonts/QuasiTimes)
```

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

The `eulervm` package virtually extends the Euler fonts.

```
4103 \DeclareMicrotypeAlias{zeur}{eur} % Euler VM
4104 \DeclareMicrotypeAlias{zeus}{eus} % "
```

MicroPress’s Charter version (`chmath`).

```
4105 \DeclareMicrotypeAlias{chr}{bch} % CH Math
```

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

```
4106 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
4107 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
4108 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

Euro symbol fonts, to save some files.

```
4109 \DeclareMicrotypeAlias{zpeus}{zpeu} % Adobe Euro sans -> serif
4110 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
4111 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
4112
```

15.3 Interaction with babel

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

```
4113 %%% -----
4114 %%% INTERACTION WITH THE `babel' PACKAGE
4115
4116 \DeclareMicrotypeBabelHook
4117   {english,UKenglish,british,USenglish,american}
4118   {kerning=, spacing=nonfrench}
4119
4120 \DeclareMicrotypeBabelHook
4121   {french,francais,acadian,canadien}
4122   {kerning=french, spacing=}
4123
4124 \DeclareMicrotypeBabelHook
4125   {turkish}
4126   {kerning=turkish, spacing=}
4127
```

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`

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.

```

4128 </m-t>
4129 <*-m-t|zpeu|mvs>
4130 %%% -----
4131 %%% CHARACTER INHERITANCE
4132
4133 </m-t|zpeu|mvs>
4134 <*-m-t>

```

```

4135 \DeclareCharacterInheritance
4136 { encoding = OT1 }
4137 { f = {011}, % ff
4138   i = {\i},
4139   j = {\j},
4140   O = {\O},
4141   o = {\o}
4142 }
4143

```

```
4144 \DeclareCharacterInheritance
4145 { encoding = T1 }
4146 { A = {\`A,\`A,\^A,\~A,\"A,\r A,\k A,\u A},
```

```

4147 a = {\`a,\`a,\^a,\~a,\"a,\r a,\k a,\u a},
4148 C = {\`C,\`c C,\v C},
4149 c = {\`c,\`c c,\v c},
4150 D = {\v D,\DH},
4151 d = {\v d,\dj},
4152 E = {\`E,\`E,\^E,\"E,\k E,\v E},
4153 e = {\`e,\`e,\^e,\"e,\k e,\v e},
4154 f = {027}, % ff
4155 G = {\u G},
4156 g = {\u g},
4157 I = {\`I,\`I,\^I,\"I,\.I},
4158 i = {\`i,\`i,\^i,\"i,\i},
4159 j = {\j},
4160 L = {\L,\`L,\v L},
4161 l = {\l,\`l,\v l},
4162 N = {\`N,\~N,\v N},
4163 n = {\`n,\~n,\v n},
4164 O = {\O,\`O,\`O,\^O,\~O,\"O,\H O},
4165 o = {\o,\`o,\`o,\^o,\~o,\"o,\H o},
4166 R = {\`R,\v R},
4167 r = {\`r,\v r},
4168 S = {\`S,\`c S,\v S,\SS},
4169 s = {\`s,\`c s,\v s},
4170 T = {\c T,\v T},
4171 t = {\c t,\v t},
4172 U = {\`U,\`U,\^U,\"U,\H U,\r U},
4173 u = {\`u,\`u,\^u,\"u,\H u,\r u},
4174 Y = {\`Y,\"Y},
4175 y = {\`y,\"y},
4176 Z = {\`Z,\.Z,\v Z},
4177 z = {\`z,\.z,\v z}

```

The ‘soft hyphen’ often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```

4178 % - = {127},
4179 }
4180

```

15.5.3 LY1

More characters: 008 (‘fl’), 012 (‘fi’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```

4181 \DeclareCharacterInheritance
4182 { encoding = LY1 }
4183 { A = {\`A,\`A,\^A,\~A,\"A,\r A},
4184   a = {\`a,\`a,\^a,\~a,\"a,\r a},
4185   C = {\c C},
4186   c = {\c c},
4187   D = {\DH},
4188   E = {\`E,\`E,\^E,\"E},
4189   e = {\`e,\`e,\^e,\"e},
4190   f = {011}, % ff
4191   I = {\`I,\`I,\^I,\"I},
4192   i = {\`i,\`i,\^i,\"i,\i},
4193   L = {\L},
4194   l = {\l},
4195   N = {\~N},
4196   n = {\~n},
4197   O = {\`O,\`O,\^O,\~O,\"O,\O},
4198   o = {\`o,\`o,\^o,\~o,\"o,\o},
4199   S = {\v S},
4200   s = {\v s},

```

```

4201     U = {\`U,\`U,\^U,\`U},
4202     u = {\`u,\`u,\^u,\`u},
4203     Y = {\`Y,\`Y},
4204     y = {\`y,\`y},
4205     Z = {\`v Z},
4206     z = {\`v z}
4207 }
4208

```

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.

```

4209 \DeclareCharacterInheritance
4210 { encoding = OT4 }
4211 { A = {\k A},
4212   a = {\k a},
4213   C = {\`C},
4214   c = {\`c},
4215   E = {\k E},
4216   e = {\k e},
4217   f = {011}, % ff
4218   i = {\i},
4219   j = {\j},
4220   L = {\L},
4221   l = {\l},
4222   N = {\`N},
4223   n = {\`n},
4224   O = {\O,\`O},
4225   o = {\o,\`o},
4226   S = {\`S},
4227   s = {\`s},
4228   Z = {\`Z,\`Z},
4229   z = {\`z,\`z}
4230 }
4231

```

15.5.5 QX

The Central European QX encoding.¹⁵ Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

4232 \DeclareCharacterInheritance
4233 { encoding = QX }
4234 { A = {\`A,\`A,\^A,\`A,\`A,\k A,\AA},
4235   a = {\`a,\`a,\^a,\`a,\`a,\k a,\aa},
4236   C = {\`C,\`C},
4237   c = {\`c,\`c},
4238   D = {\DH},
4239   E = {\`E,\`E,\^E,\`E,\k E},
4240   e = {\`e,\`e,\^e,\`e,\k e},
4241   f = {011}, % ff
4242   I = {\`I,\`I,\^I,\`I,\k I},
4243   i = {\`i,\`i,\^i,\`i,\k i,\i},
4244   j = {\j},
4245   L = {\L},
4246   l = {\l},
4247   N = {\`N,\`N},

```

15 Contributed by Maciej Eder.

```

4248 n = {\n,\~n},
4249 o = {\0,\`0,\'0,\^0,\-0,\"0},
4250 o = {\o,\`o,\'o,\^o,\-o,\"o},

```

The Rumanian `\textcommabelow` accents are actually replacements for the `\c` variants, which had previously (and erroneously¹⁶) been included in QX encoding. They are still kept for backwards compatibility.

```

4251 S = {\S,\c S,\textcommabelow S,\v S},
4252 s = {\s,\c s,\textcommabelow s,\v s},
4253 T = {\c T,\textcommabelow T},
4254 t = {\c t,\textcommabelow t},
4255 U = {\`U,\'U,\^U,\"U,\k U},
4256 u = {\`u,\'u,\^u,\"u,\k u},
4257 Y = {\`Y,\"Y},
4258 y = {\`y,\"y},
4259 Z = {\`Z,\.Z,\v Z},
4260 z = {\`z,\.z,\v z},
4261 . = \textellipsis
4262 }
4263

```

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.

```

4264 \DeclareCharacterInheritance
4265 { encoding = T5 }
4266 { A = {\`A,\'A,\-A,\h A,\d A,\^A,\u A,
4267       \`Acircumflex,\'Acircumflex,\-Acircumflex,\hAcircumflex,\dAcircumflex,
4268       \`Abreve,\'Abreve,\-Abreve,\hAbreve,\dAbreve},
4269 a = {\`a,\'a,\-a,\h a,\d a,\^a,\u a,
4270       \`acircumflex,\'acircumflex,\-acircumflex,\hacircumflex,\dacircumflex,
4271       \`abreve,\'abreve,\-abreve,\habreve,\dabreve},
4272 D = {\DJ},
4273 d = {\dj},
4274 E = {\`E,\'E,\-E,\h E,\d E,\^E,
4275       \`Ecircumflex,\'Ecircumflex,\-Ecircumflex,\hEcircumflex,\dEcircumflex},
4276 e = {\`e,\'e,\-e,\h e,\d e,\^e,
4277       \`ecircumflex,\'ecircumflex,\-ecircumflex,\hecircumflex,\decircumflex},
4278 I = {\`I,\'I,\-I,\h I,\d I},
4279 i = {\`i,\'i,\-i,\h i,\d i,\i},
4280 O = {\`O,\'O,\-O,\h O,\d O,\^O,\horn O,
4281       \`Ocircumflex,\'Ocircumflex,\-Ocircumflex,\hOcircumflex,\dOcircumflex,
4282       \`Ohorn,\'Ohorn,\-Ohorn,\hOhorn,\dOhorn},
4283 o = {\`o,\'o,\-o,\h o,\d o,\^o,\horn o,
4284       \`ocircumflex,\'ocircumflex,\-ocircumflex,\hocircumflex,\docircumflex,
4285       \`ohorn,\'ohorn,\-ohorn,\hohorn,\dohorn},
4286 U = {\`U,\'U,\-U,\h U,\d U,\^U,\horn U,
4287       \`Uhorn,\'Uhorn,\-Uhorn,\hUhorn,\dUhorn},
4288 u = {\`u,\'u,\-u,\h u,\d u,\^u,\horn u,
4289       \`uhorn,\'uhorn,\-uhorn,\huhorn,\duhorn},
4290 Y = {\`Y,\'Y,\-Y,\h Y,\d Y},
4291 y = {\`y,\'y,\-y,\h y,\d y}
4292 }
4293
4294 /m-t

```

16 Cf. <http://tug.org/pipermail/tex-live/2008-August/017204.html>

15.5.7 Euro symbols

Make Euro symbols settings simpler.

```

4295 < *zpeu >
4296 \DeclareCharacterInheritance
4297   { encoding = U,
4298     family   = {zpeu,zpeus,eurosans} }
4299   { E = 128 }
4300
4301 < /zpeu >
4302 < *mvs >
4303 \DeclareCharacterInheritance
4304   { encoding = OT1,
4305     family   = mvs }
4306   { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
4307

```

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.

```

4308 \DeclareCharacterInheritance
4309   { encoding = U,
4310     family   = mvs }
4311   { 164 = {099,100,101} }
4312
4313 < /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.

```

4314 < *m-t >
4315 %%% -----
4316 %%% TRACKING/LETTERSPACING
4317
4318 \SetTracking
4319   [ name       = default,
4320     no ligatures = {f} ]
4321   { encoding    = {OT1,T1,T2A,Ly1,OT4,QX} }
4322   { }
4323

```

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).

```

4324 %%% -----
4325 %%% EXPANSION
4326
4327 \SetExpansion
4328   [ name       = default      ]
4329   { encoding    = {OT1,OT4,QX,T1,Ly1} }
4330   {
4331     A = 500,      a = 700,
4332     \AE = 500,    \ae = 700,
4333     B = 700,      b = 700,
4334     C = 700,      c = 700,
4335     D = 500,      d = 700,

```

```

4336     E = 700,      e = 700,
4337     F = 700,
4338     G = 500,      g = 700,
4339     H = 700,      h = 700,
4340     K = 700,      k = 700,
4341     M = 700,      m = 700,
4342     N = 700,      n = 700,
4343     O = 500,      o = 700,
4344     \OE = 500,    \oe = 700,
4345     P = 700,      p = 700,
4346     Q = 500,      q = 700,
4347     R = 700,
4348     S = 700,      s = 700,
4349     U = 700,      u = 700,
4350     W = 700,      w = 700,
4351     Z = 700,      z = 700,
4352     2 = 700,
4353     3 = 700,
4354     6 = 700,
4355     8 = 700,
4356     9 = 700
4357 }
4358

```

Settings for Cyrillic T2A encoding.¹⁷

```

4359 \SetExpansion
4360 [ name      = T2A ]
4361 { encoding = T2A }
4362 {
4363     A = 500,      a = 700,
4364     B = 700,      b = 700,
4365     C = 700,      c = 700,
4366     D = 500,      d = 700,
4367     E = 700,      e = 700,
4368     F = 700,
4369     G = 500,      g = 700,
4370     H = 700,      h = 700,
4371     K = 700,      k = 700,
4372     M = 700,      m = 700,
4373     N = 700,      n = 700,
4374     O = 500,      o = 700,
4375     P = 700,      p = 700,
4376     Q = 500,      q = 700,
4377     R = 700,
4378     S = 700,      s = 700,
4379     U = 700,      u = 700,
4380     W = 700,      w = 700,
4381     Z = 700,      z = 700,
4382     2 = 700,
4383     3 = 700,
4384     6 = 700,
4385     8 = 700,
4386     9 = 700,
4387     \CYRA = 500,    \cyra = 700,
4388     \CYRB = 700,    \cyrb = 700,
4389     \CYRV = 700,    \cyrv = 700,
4390     \CYRG = 700,    \cyrg = 700,
4391     \CYRD = 700,    \cyrd = 700,
4392     \CYRE = 700,    \cyre = 700,
4393     \CYRZH = 700,   \cyrzh = 700,
4394     \CYRZ = 700,    \cyrz = 700,

```

¹⁷ Contributed by *Karl Karlsson*.


```

4395     \CYRI = 700,      \cyri = 700,
4396     \CYRISHRT = 700, \cyrishrt = 700,
4397     \CYRK = 700,      \cyrk = 700,
4398     \CYRL = 700,      \cyr1 = 700,
4399     \CYRM = 700,      \cym = 700,
4400     \CYRN = 700,      \cyrn = 700,
4401     \CYRO = 500,      \cyro = 700,
4402     \CYRP = 700,      \cyrp = 700,
4403     \CYRR = 700,      \cyr = 700,
4404     \CYRS = 700,      \cyr = 700,
4405     \CYRT = 700,      \cyr = 700,
4406     \CYRU = 700,      \cyr = 700,
4407     \CYRF = 700,      \cyr = 700,
4408     \CYRH = 700,      \cyr = 700,
4409     \CYRC = 700,      \cyr = 700,
4410     \CYRCH = 700,     \cyrch = 700,
4411     \CYRSH = 700,     \cyrsh = 700,
4412     \CYRSHCH = 700,   \cyrshch = 700,
4413     \CYRHRSN = 700,   \cyrhrdsn = 700,
4414     \CYRERY = 700,    \cyrery = 700,
4415     \CYRSFTSN = 700,  \cyrfts = 700,
4416     \CYREREV = 700,   \cyrerev = 700,
4417     \CYRYU = 700,     \cyr = 700,
4418     \CYRYA = 700,     \cyr = 700
4419   }
4420

```

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

```

4421 \SetExpansion
4422 [ name = T5 ]
4423 { encoding = T5 }
4424 {
4425     A = 500,      a = 700,
4426     B = 700,      b = 700,
4427     C = 700,      c = 700,
4428     D = 500,      d = 700,
4429     E = 700,      e = 700,
4430     F = 700,
4431     G = 500,      g = 700,
4432     H = 700,      h = 700,
4433     K = 700,      k = 700,
4434     M = 700,      m = 700,
4435     N = 700,      n = 700,
4436     O = 500,      o = 700,
4437     P = 700,      p = 700,
4438     Q = 500,      q = 700,
4439     R = 700,
4440     S = 700,      s = 700,
4441     U = 700,      u = 700,
4442     W = 700,      w = 700,
4443     Z = 700,      z = 700,
4444     2 = 700,
4445     3 = 700,
4446     6 = 700,
4447     8 = 700,
4448     9 = 700
4449   }
4450
4451 \m-t

```

15.8 Character protrusion

```

4452 %%% -----
4453 %%% PROTRUSION
4454

```

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},
  \textquoteright  = {700, },    \textquoteright  = { ,700},
  \textquotedblleft = {500, },    \textquotedblright = { ,500}
}

```

15.8.1 Normal

The default settings always use the most moderate value.

```

4455 <*cfg-t>
4456 \SetProtrusion
4457 <m-t> [ name      = default ]

```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code `bch`)

```

4458 <bch> [ name      = bch-default ]

```

- Bitstream Letter Gothic (`blg`)

```

4459 <blg> [ name      = blg-default ]

```

- Computer Modern Roman (`cmr`)

```

4460 <cmr> [ name      = cmr-default ]

```

- Adobe Garamond (`pad`, `padx`, `padj`)

```

4461 <pad> [ name      = pad-default ]

```

- Minion¹⁸ (pmnx, pmnj)

```
4462 <pmn> [ name = pmnj-default ]
```

- Palatino (ppl, pplx, pplj)

```
4463 <ppl> [ name = ppl-default ]
```

- Times (ptm, ptmx, ptmj)

```
4464 <ptm> [ name = ptm-default ]
```

- URW Garamond (ugm)

```
4465 <ugm> [ name = ugm-default ]
4466 <m-t|cmr> { }
4467 <bch|blg|pad|pmn|ugm> { encoding = OT1,
4468 <ppl|ptm> { encoding = {OT1,OT4},
4469 <bch> family = bch }
4470 <blg> family = blg }
4471 <pad> family = {pad,padx,padj }
4472 <pmn> family = pmnj }
4473 <ppl> family = {ppl,pplx,pplj }
4474 <ptm> family = {ptm,ptmx,ptmj }
4475 <ugm> family = ugm }
4476 {
4477 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> A = {50,50},
4478 <ugm> A = {50,100},
4479 <pad|ptm> \AE = {50, },
4480 <ugm> \AE = {150,50},
4481 <ugm> B = { ,50},
4482 <bch|pad|pmn|ugm> C = {50, },
4483 <bch|pad|pmn> D = { ,50},
4484 <ugm> D = { ,70},
4485 <ugm> E = { ,50},
4486 <m-t|bch|cmr|pad|pmn|ptm> F = { ,50},
4487 <ugm> F = { ,70},
4488 <bch|pad|pmn> G = {50, },
4489 <ugm> G = {50,50},
4490 <blg> I = {150,150},
4491 <m-t|cmr|pad|pmn|ppl|ptm|ugm> J = {50, },
4492 <bch|blg> J = {100, },
4493 <!blg> K = { ,50},
4494 <blg> K = {50, },
4495 <m-t|bch|cmr|pad|pmn|ppl> L = { ,50},
4496 <blg> L = { ,150},
4497 <ptm> L = { ,80},
4498 <ugm> L = { ,120},
4499 <bch|pad|pmn|ugm> O = {50,50},
4500 <pad|pmn> \OE = {50, },
4501 <ugm> \OE = {50,50},
4502 <blg> P = { ,100},
4503 <ugm> P = { ,50},
4504 <bch|pad|pmn> Q = {50,70},
4505 <ugm> Q = {50,50},
4506 <bch> R = { ,50},
4507 <ugm> R = { ,70},
4508 <m-t|bch|cmr|pad|pmn|ppl|ptm> T = {50,50},
4509 <blg> T = {100,100},
4510 <ugm> T = {70,70},
4511 <m-t|bch|cmr|pad|pmn|ppl|ptm> V = {50,50},
4512 <blg|ugm> V = {70,70},
```

18 Contributed by Harald Harders.

```

4513 <m-t|bch|cmr|pad|pmn|ppl|ptm>    W = {50,50},
4514 <ugm>    W = {70,70},
4515 <m-t|bch|cmr|pad|pmn|ppl|ptm>    X = {50,50},
4516 <ugm>    X = {50,70},
4517 <m-t|bch|cmr|pad|pmn|ppl>    Y = {50,50},
4518 <blg|ptm|ugm>    Y = {80,80},
4519 <ugm>    Z = {50,50},
4520 <blg>    f = {150,100},
4521 <blg>    i = {150,150},
4522 <blg>    j = {100,100},
4523 <m-t|bch|cmr|pad|pmn|ppl|ptm>    k = { ,50},
4524 <ugm>    k = { ,70},
4525 <blg>    l = {150,150},
4526 <pmn>    l = { ,50},
4527 <pad|ppl>    p = {50,50},
4528 <ugm>    p = { ,50},
4529 <pad|ppl>    q = {50, },
4530 <!blg>    r = { ,50},
4531 <blg>    r = {100, 80},
4532 <cmr|pad|pmn>    t = { ,70},
4533 <bch>    t = { ,50},
4534 <blg>    t = {150, 80},
4535 <ugm>    t = { ,100},
4536 <m-t|bch|cmr|pad|pmn|ppl|ptm>    v = {50,50},
4537 <blg>    v = {100,100},
4538 <ugm>    v = {50,70},
4539 <m-t|bch|cmr|pad|pmn|ppl|ptm>    w = {50,50},
4540 <ugm>    w = {50,70},
4541 <!blg>    x = {50,50},
4542 <blg>    x = {100,100},
4543 <m-t|bch|pad|pmn>    y = { ,50},
4544 <blg>    y = { 50,100},
4545 <cmr|ppl|ptm>    y = {50,70},
4546 <ugm>    y = { ,70},

4547 <cmr>    0 = { ,50},
4548 <m-t>    1 = {50,50},
4549 <bch|blg|pad|ptm|ugm>    1 = {150,150},
4550 <cmr>    1 = {100,200},
4551 <pmn>    1 = { ,50},
4552 <ppl>    1 = {100,100},
4553 <bch|cmr|pad|ugm>    2 = {50,50},
4554 <blg>    2 = { ,100},
4555 <bch|pmn>    3 = {50, },
4556 <cmr|pad|ugm>    3 = {50,50},
4557 <blg>    3 = {100, },
4558 <m-t|pad>    4 = {50,50},
4559 <bch>    4 = {100,50},
4560 <blg>    4 = {100, },
4561 <cmr|ugm>    4 = {70,70},
4562 <pmn>    4 = {50, },
4563 <ptm>    4 = {70, },
4564 <cmr>    5 = { ,50},
4565 <pad>    5 = {50,50},
4566 <bch>    6 = {50, },
4567 <cmr>    6 = { ,50},
4568 <pad>    6 = {50,50},
4569 <m-t>    7 = {50,50},
4570 <bch|pad|pmn|ugm>    7 = {50,80},
4571 <blg>    7 = {100,100},
4572 <cmr|ptm>    7 = {50,100},
4573 <ppl>    7 = { ,50},
4574 <cmr>    8 = { ,50},

```

```

4575 <bch|pad> 9 = {50,50},
4576 <cmr> 9 = { ,50},
4577 <m-t|cmr|pad|pmn|ppl|ptm|ugm> . = { ,700},
4578 <bch> . = { ,600},
4579 <blg> . = {400,500},
4580 <!blg> {,}= { ,500},
4581 <blg> {,}= {300,400},
4582 <m-t|cmr|pad|pmn|ppl|ptm|ugm> : = { ,500},
4583 <bch> : = { ,400},
4584 <blg> : = {300,400},
4585 <m-t|bch|pad|pmn|ptm> ; = { ,300},
4586 <blg> ; = {200,300},
4587 <cmr|ppl> ; = { ,500},
4588 <ugm> ; = { ,400},
4589 <!blg> ! = { ,100},
4590 <blg> ! = {200,200},
4591 <m-t|pad|pmn|ptm> ? = { ,100},
4592 <bch|cmr|ppl|ugm> ? = { ,200},
4593 <blg> ? = {150,150},
4594 <pmn> " = {300,300},
4595 <m-t|bch|cmr|pad|pmn|ppl> @ = {50,50},
4596 <ptm> @ = {100,100},
4597 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> ~ = {200,250},
4598 <ugm> ~ = {300,350},
4599 <pad|ppl|ptm> & = {50,100},
4600 <ugm> & = { ,100},
4601 <m-t|cmr|pad|pmn> \% = {50,50},
4602 <bch> \% = { ,50},
4603 <ppl|ptm> \% = {100,100},
4604 <ugm> \% = {50,100},
4605 <blg> \# = {100,100},
4606 <m-t|ppl|ptm|ugm> * = {200,200},
4607 <bch|pmn> * = {200,300},
4608 <blg> * = {150,200},
4609 <cmr|pad> * = {300,300},
4610 <m-t|cmr|ppl|ptm> + = {250,250},
4611 <bch> + = {150,250},
4612 <pad> + = {300,300},
4613 <blg|pmn> + = {150,200},
4614 <ugm> + = {250,300},
4615 <blg|ugm> {=}= {200,200},
4616 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
4617 <bch|ugm> ( = {200, }, ) = { ,200},
4618 <cmr|blg> ( = {300, }, ) = { ,300},
4619 <ppl> ( = {100, }, ) = { ,300},
4620 <bch|pmn> [ = {100, }, ] = { ,100},
4621 <blg> [ = {300,100}, ] = { ,300},

4622 <m-t|pad|pmn|ptm> / = {100,200},
4623 <bch> / = { ,200},
4624 <blg> / = {300,300},
4625 <cmr|ppl> / = {200,300},
4626 <ugm> / = {100,300},
4627 <m-t|ptm> - = {500,500},
4628 <bch|cmr|ppl> - = {400,500},
4629 <blg> - = {300,400},
4630 <pad> - = {300,500},
4631 <pmn> - = {200,400},
4632 <ugm> - = {500,600},
4633 <blg> < = {200,100}, > = {100,200},
4634 <blg> _ = {150,250},
4635 <blg> | = {250,250},
4636 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},

```

```

4637 <bch> \textendash = {200,300}, \textemdash = {150,250},
4638 <cmr> \textendash = {400,300}, \textemdash = {300,200},
4639 <pad|ppl|ptm> \textendash = {300,300}, \textemdash = {200,200},
4640 <ugm> \textendash = {250,300}, \textemdash = {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).

```

4641 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
4642 <blg> \textquoteleft = {400,600}, \textquoteright = {400,600},
4643 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
4644 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
4645 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
4646 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
4647 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300}
4648 <blg> \textquotedblright = {300,400}
4649 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600}
4650 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4651 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400}
4652 }
4653

```

Greek uppercase letters are in OT1 encoding only.

```

4654 <*m-t|cmr>
4655 \SetProtrusion
4656 <m-t> [ name = OT1-default,
4657 <cmr> [ name = cmr-OT1,
4658 <m-t> load = default ]
4659 <cmr> load = cmr-default ]
4660 <m-t> { encoding = OT1 }
4661 <cmr> { encoding = {OT1,OT4},
4662 <cmr> family = cmr }
4663 {
4664 \AE = {50, },
4665 <*cmr>
4666 "00 = { ,150}, % \Gamma
4667 "01 = {100,100}, % \Delta
4668 "02 = { 50, 50}, % \Theta
4669 "03 = {100,100}, % \Lambda
4670 "06 = { 50, 50}, % \Sigma
4671 "07 = {100,100}, % \Upsilon
4672 "08 = { 50, 50}, % \Phi
4673 "09 = { 50, 50} % \Psi

```

Remaining slots can be found in the source file.

```

4674 </cmr>
4675 }
4676
4677 </m-t|cmr>

```

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

```

4678 \SetProtrusion
4679 <m-t> [ name = T1-default,
4680 <bch> [ name = bch-T1,
4681 <blg> [ name = blg-T1,
4682 <cmr> [ name = cmr-T1,
4683 <pad> [ name = pad-T1,
4684 <pmn> [ name = pmnj-T1,
4685 <ppl> [ name = ppl-T1,
4686 <ptm> [ name = ptm-T1,
4687 <ugm> [ name = ugm-T1,
4688 <m-t> load = default ]

```

```

4689 <bch>      load      = bch-default ]
4690 <blg>      load      = blg-default ]
4691 <cmr>      load      = cmr-default ]
4692 <pad>      load      = pad-default ]
4693 <pmn>      load      = pmnj-default ]
4694 <ppl>      load      = ppl-default ]
4695 <ptm>      load      = ptm-default ]
4696 <ugm>      load      = ugm-default ]
4697 <m-t>      { encoding = {T1,LY1}      }
4698 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4699 <blg|ptm|ugm>      { encoding = {T1},
4700 <bch>      family    = bch }
4701 <blg>      family    = blg }
4702 <cmr>      family    = cmr }
4703 <pad>      family    = {pad,padx,padj} }
4704 <pmn>      family    = pmnj }
4705 <ppl>      family    = {ppl,pplx,pplj} }
4706 <ptm>      family    = {ptm,ptmx,ptmj} }
4707 <ugm>      family    = ugm }
4708 {
4709 <m-t|cmr>      \AE = {50, },
4710 <bch>      \OE = {50, },
4711 <pmn>      \TH = { ,50},
4712 <blg>      \v L = { ,250},
4713 <blg>      \v d = { ,250},
4714 <blg>      \v l = { ,250},
4715 <blg>      \v t = { ,250},
4716 <blg>      127 = {300,400},
4717 <blg>      156 = {100, }, % IJ
4718 <blg>      188 = { 80, 80}, % ij
4719 <m-t|bch|pad|pmn|ppl|ptm>      _ = {100,100},
4720 <cmr>      _ = {200,200},
4721 <ugm>      _ = {100,200},
4722 <m-t|pad|pmn|ptm>      \textbackslash = {100,200},
4723 <bch>      \textbackslash = {150,200},
4724 <blg>      \textbackslash = {250,300},
4725 <cmr|ppl>      \textbackslash = {200,300},
4726 <ugm>      \textbackslash = {100,300},
4727 <ugm>      \textbar = {200,200},
4728 <blg>      \textendash = {300,300}, \textemdash = {150,150},
4729 <blg>      \textquotedbl = {300,400}, \textquotedblleft = {300,400},
4730 <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.

```

4731 <m-t|cmr|pad|ppl|ptm|ugm>      \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4732 <blg>      \quotesinglbase = {400,400}, \quotedblbase = {300,400},
4733 <bch|pmn>      \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4734 <m-t|bch|pmn>      \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4735 <blg>      \guilsinglleft = {300,500}, \guilsinglright = {300,500},
4736 <cmr|pad|ppl|ptm>      \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4737 <ugm>      \guilsinglleft = {400,400}, \guilsinglright = {300,600},
4738 <m-t>      \guillemotleft = {200,200}, \guillemotright = {200,200},
4739 <cmr>      \guillemotleft = {300,200}, \guillemotright = {100,400},
4740 <bch|pmn>      \guillemotleft = {200,200}, \guillemotright = {150,300},
4741 <blg|pad|ppl|ptm>      \guillemotleft = {300,300}, \guillemotright = {200,400},
4742 <ugm>      \guillemotleft = {300,400}, \guillemotright = {300,400},
4743 <m-t|bch|cmr|pad|pmn|ppl|ugm>      \textexclamdown = {100, }, \textquestiondown = {100, },
4744 <blg>      \textexclamdown = {200, }, \textquestiondown = {100, },
4745 <ptm>      \textexclamdown = {200, }, \textquestiondown = {200, },
4746 <m-t|cmr|pad|ppl|ptm|ugm>      \textbraceleft = {400,200}, \textbraceright = {200,400},
4747 <bch|blg|pmn>      \textbraceleft = {200, }, \textbraceright = { ,300},

```

```

4748 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
4749 <pmn> \textless = {100, }, \textgreater = { ,100},
4750 <pmn> \textvisiblespace = {100,100} % not in LY1
4751 }
4752

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```

4753 <*cmr>
4754 \SetProtrusion
4755 [ name = lmr-T1,
4756   load = cmr-T1 ]
4757 { encoding = {T1,LY1},
4758   family = lmr }
4759 {
4760   \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4761 }
4762
4763 </cmr>

```

Settings for the T2A encoding (generic and Computer Modern Roman).¹⁹

```

4764 <*m-t|cmr>
4765 \SetProtrusion
4766 <m-t> [ name = T2A-default,
4767 <cmr> [ name = cmr-T2A,
4768 <m-t>   load = default ]
4769 <cmr>   load = cmr-default ]
4770 { encoding = T2A,
4771 <m-t> }
4772 <cmr> family = cmr }
4773 {
4774   \CYRA = {50,50},
4775   \CYRG = { ,50},
4776   \CYRK = { ,50},
4777   \CYRT = {50,50},
4778   \CYRH = {50,50},
4779   \CYRU = {50,50},
4780   \cyrk = { ,50},
4781   \cyrh = { ,50},
4782   \cyrh = {50,50},
4783 <m-t>   \cyru = {50,50},
4784 <cmr>   \cyru = {50,70},
4785 <m-t>   - = {100,100},
4786 <cmr>   - = {200,200},
4787 <m-t>   \textbackslash = {100,200}, \quotedblbase = {400,400},
4788 <cmr>   \textbackslash = {200,300}, \quotedblbase = {400,400},
4789 <cmr>   \textquotedbl = {300,300}, \textquotedblleft = {200,600},
4790 <m-t>   \guillemotleft = {200,200}, \guillemotright = {200,200},
4791 <cmr>   \guillemotleft = {300,200}, \guillemotright = {100,400},
4792   \textbraceleft = {400,200}, \textbraceright = {200,400},
4793   \textless = {200,100}, \textgreater = {100,200}
4794 }
4795
4796 </m-t|cmr>

```

Settings for the QX encoding (generic and Times).²⁰ It also includes some glyphs otherwise in TS1.

```

4797 <*m-t|ptm>
4798 \SetProtrusion

```

19 Contributed by Karl Karlsson.

20 Contributed by Maciej Eder.


```

4799 <m-t> [ name      = QX-default,
4800 <ptm> [ name      = ptm-QX,
4801 <m-t>   load      = default ]
4802 <ptm>   load      = ptm-default ]
4803 <m-t>   { encoding = QX }
4804 <ptm>   { encoding = QX,
4805 <ptm>     family   = {ptm,ptmx,ptmj} }
4806   {
4807     \AE = {50, },
4808 <ptm>   * = {200,200},
4809     {=} = {100,100},
4810     \textunderscore = {100,100},
4811     \textbackslash = {100,200},
4812     \quotedblbase = {400,400},
4813 <m-t>     \guillemotleft = {200,200}, \guillemotright = {200,200},
4814 <ptm>     \guillemotleft = {300,300}, \guillemotright = {200,400},
4815     \textexclamdown = {100, }, \textquestiondown = {100, },
4816 <m-t>     \textbraceleft = {400,200}, \textbraceright = {200,400},
4817 <ptm>     \textbraceleft = {200,200}, \textbraceright = {200,300},
4818     \textless = {200,100}, \textgreater = {100,200},
4819     \textminus = {200,200}, \textdegree = {300,300},
4820 <m-t>     \copyright = {100,100}, \textregistered = {100,100}
4821 <ptm>     \copyright = {100,150}, \textregistered = {100,150},
4822 <ptm>     \textxgeq = { ,100}, \textxleq = {100, },
4823 <ptm>     \textalpha = { , 50}, \textDelta = { 70, 70},
4824 <ptm>     \textpi = { 50, 80}, \textSigma = { , 70},
4825 <ptm>     \textmu = { , 80}, \texteuro = { 50, 50},
4826 <ptm>     \textellipsis = {150,200}, \textasciitilde = { 80, 80},
4827 <ptm>     \textapprox = { 50, 50}, \textinfty = {100,100},
4828 <ptm>     \textdagger = {150,150}, \textdaggerdbl = {100,100},
4829 <ptm>     \textdiv = { 50,150}, \textsection = { 80, 80},
4830 <ptm>     \texttimes = {100,150}, \textpm = { 50, 80},
4831 <ptm>     \textbullet = {150,150}, \textperiodcentered = {300,300},
4832 <ptm>     \textquotesingle = {500,500}, \textquotedbl = {300,300},
4833 <ptm>     \textperthousand = { ,50}
4834   }
4835
4836 </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.

```

4837 <*cmr|bch>
4838 \SetProtrusion
4839 <cmr> [ name      = cmr-T5,
4840 <cmr>   load      = cmr-default ]
4841 <bch> [ name      = bch-T5,
4842 <bch>   load      = bch-default ]
4843   { encoding = T5,
4844 <cmr>     family   = cmr }
4845 <bch>     family   = bch }
4846   {
4847 <bch>     _ = {100,100},
4848 <bch>     \textbackslash = {150,200},
4849 <cmr>     \textbackslash = {200,300},
4850 <cmr>     \textquotedblleft = {200,600},
4851 <cmr>     \textquotedbl = {300,300},
4852 <bch>     \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4853 <cmr>     \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4854 <bch>     \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4855 <cmr>     \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4856 <bch>     \guillemotleft = {200,200}, \guillemotright = {150,300},
4857 <cmr>     \guillemotleft = {300,200}, \guillemotright = {100,400},

```

```

4858 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
4859 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
4860 \textless = {200,100}, \textgreater = {100,200}
4861 }
4862
4863 </cmr><bch>
4864 <*pmn>
4865 \SetProtrusion
4866 [ name = pmnx-OT1,
4867 load = pmnj-default ]
4868 { encoding = OT1,
4869 family = pmnx }
4870 {
4871 1 = {230,180}
4872 }
4873
4874 \SetProtrusion
4875 [ name = pmnx-T1,
4876 load = pmnj-T1 ]
4877 { encoding = {T1,LY1},
4878 family = pmnx }
4879 {
4880 1 = {230,180}
4881 }
4882
4883 </pmn>

```

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

```

4884 <*ptm>
4885 \SetProtrusion
4886 [ name = ptm-LY1,
4887 load = ptm-T1 ]
4888 { encoding = LY1,
4889 family = {ptm,ptmx,ptmj} }
4890 {
4891 - = {100,100},
4892 \texttrademark = {100,100},
4893 \textregistered = {100,100},
4894 \textcopyright = {100,100},
4895 \textdegree = {300,300},
4896 \textminus = {200,200},
4897 \textellipsis = {150,200},
4898 % \texteuro = { , }, % ?
4899 \textcent = {100,100},
4900 \textquotesingle = {500,500},
4901 \textflorin = { 50, 70},
4902 \textdagger = {150,150},
4903 \textdaggerdbl = {100,100},
4904 \textperthousand = { , 50},
4905 \textbullet = {150,150},
4906 \textonesuperior = {100,100},
4907 \texttwosuperior = { 50, 50},
4908 \textthreesuperior = { 50, 50},
4909 \textperiodcentered = {300,300},
4910 \textplusminus = { 50, 80},
4911 \textmultiply = {100,100},
4912 \textdivide = { 50,150}

```

Remaining slots in the source file.

```

4913 }
4914

```

4915 *</ptm>*

15.8.2 Italics

To find default settings for italic is difficult, since the character shapes and their behaviour at the beginning or end of line may be wildly different for different fonts. Therefore, we leave the letters away, and only set up the punctuation characters.

```

4916 \SetProtrusion
4917 <m-t> [ name = OT1-it ]
4918 <bch> [ name = bch-it ]
4919 <blg> [ name = blg-it,
4920 <blg> load = blg-default ]
4921 <cmr> [ name = cmr-it ]
4922 <pad> [ name = pad-it ]
4923 <pmn> [ name = pmn-it ]
4924 <ppl> [ name = ppl-it ]
4925 <ptm> [ name = ptm-it ]
4926 <ugm> [ name = ugm-it ]
4927 <m-t|bch|blg|pad|pmn|ugm> { encoding = OT1,
4928 <ppl|ptm> { encoding = {OT1,OT4},
4929 <bch> family = bch,
4930 <blg> family = blg,
4931 <pad> family = {pad,padx,padj},
4932 <pmn> family = pmn,
4933 <ppl> family = {ppl,pplx,pplj},
4934 <ptm> family = {ptm,ptmx,ptmj},
4935 <ugm> family = ugm,
4936 <m-t|bch|pad|pmn|ppl|ptm> shape = {it,sl} }
4937 <blg|ugm> shape = it }
4938 <cmr> { }
4939 {
4940 <cmr|ptm> A = {100,50},
4941 <pad|pmn> A = {50, },
4942 <ugm> A = { ,150},
4943 <ppl> A = {50,50},
4944 <ptm> \AE = {100, },
4945 <pad|ppl> \AE = {50, },
4946 <pmn> \AE = { , -50},
4947 <cmr|pad|ppl|ptm> B = {50, },
4948 <pmn> B = {20,-50},
4949 <bch|ppl|ptm|ugm> C = {50, },
4950 <cmr|pad> C = {100, },
4951 <pmn> C = {50,-50},
4952 <cmr|pad|ppl|ptm> D = {50,50},
4953 <pmn> D = {20, },
4954 <cmr|pad|ppl|ptm> E = {50, },
4955 <pmn> E = {20,-50},
4956 <cmr|pad|ptm> F = {100, },
4957 <pmn> F = {10, },
4958 <ppl> F = {50, },
4959 <bch|ppl|ptm|ugm> G = {50, },
4960 <cmr|pad> G = {100, },
4961 <pmn> G = {50,-50},
4962 <cmr|pad|ppl|ptm> H = {50, },
4963 <cmr|pad|ptm> I = {50, },
4964 <pmn> I = {20,-50},
4965 <cmr|ptm> J = {100, },
4966 <pad> J = {50, },
4967 <pmn> J = {20, },
4968 <cmr|pad|ppl|ptm> K = {50, },
4969 <pmn> K = {20, },

```

```

4970 <cmr|pad|ppl|ptm>    L = {50, },
4971 <pmn>                L = {20,50},
4972 <ugm>                L = { ,100},
4973 <cmr|ptm>           M = {50, },
4974 <pmn>                M = { , -30},
4975 <cmr|ptm>           N = {50, },
4976 <pmn>                N = { , -30},
4977 <bch|pmn|ppl|ptm>    O = {50, },
4978 <cmr|pad>           O = {100, },
4979 <ugm>                O = {70,50},
4980 <pmn|ppl|ptm>       \OE = {50, },
4981 <pad>                \OE = {100, },
4982 <cmr|pad|ppl|ptm>   P = {50, },
4983 <pmn>                P = {20,-50},
4984 <bch|pmn|ppl|ptm>   Q = {50, },
4985 <cmr|pad>           Q = {100, },
4986 <ugm>                Q = {70,50},
4987 <cmr|pad|ppl|ptm>   R = {50, },
4988 <pmn>                R = {20, },
4989 <bch|cmr|pad|ppl|ptm> S = {50, },
4990 <pmn>                S = {20,-30},
4991 <bch|cmr|pad|ppl|ptm> $ = {50, },
4992 <pmn>                $ = {20,-30},
4993 <bch|pmn|ugm>        T = {70, },
4994 <cmr|pad|ppl|ptm>   T = {100, },
4995 <cmr|pad|ppl|ptm>   U = {50, },
4996 <pmn>                U = {50,-50},
4997 <cmr|pad|pmn|ugm>   V = {100, },
4998 <ppl|ptm>           V = {100,50},
4999 <cmr|pad|pmn|ugm>   W = {100, },
5000 <ppl>               W = {50, },
5001 <ptm>               W = {100,50},
5002 <cmr|ppl|ptm>      X = {50, },
5003 <cmr|ptm>          Y = {100, },
5004 <pmn>               Y = {50, },
5005 <ppl>               Y = {100,50},
5006 <pmn>               Z = { , -50},
5007 <pmn>               d = { , -50},
5008 <pad|pmn>          f = { , -100},
5009 <pmn>               i = { , -30},
5010 <pmn>               j = { , -30},
5011 <pmn>               l = { , -100},
5012 <bch>               o = {50,50},
5013 <bch>               p = { , 50},
5014 <pmn>               p = {-50, },
5015 <bch>               q = {50, },
5016 <pmn>               r = { , 50},
5017 <bch>               t = { , 50},
5018 <pmn|ugm>          v = {50, },
5019 <bch>               w = { , 50},
5020 <pmn|ugm>          w = {50, },
5021 <bch>               y = { , 50},
5022 <cmr>               0 = {100, },
5023 <bch|ptm>           1 = {150,100},
5024 <cmr>               1 = {200,50},
5025 <pad>               1 = {150, },
5026 <pmn>               1 = {50, },
5027 <ppl>               1 = {100, },
5028 <ugm>               1 = {150,150},
5029 <cmr>               2 = {100,-100},
5030 <pad|ppl|ptm>      2 = {50, },
5031 <pmn>               2 = {-50, },
5032 <bch>               3 = {50, },

```

```

5033 <cmr>      3 = {100,-100},
5034 <pmn>      3 = {-100, },
5035 <ptm>      3 = {100,50},
5036 <bch>      4 = {100, },
5037 <cmr|pad>   4 = {150, },
5038 <ppl|ptm>   4 = {50, },
5039 <cmr>      5 = {100, },
5040 <ptm>      5 = {50, },
5041 <bch>      6 = {50, },
5042 <cmr>      6 = {100, },
5043 <bch|pad|ptm> 7 = {100, },
5044 <cmr>      7 = {200,-150},
5045 <pmn>      7 = {20, },
5046 <ppl>      7 = {50, },
5047 <cmr>      8 = {50,-50},
5048 <cmr>      9 = {100,-100},
5049 <m-t|cmr|pad|pmn|ppl> . = { ,500},
5050 <blg>      . = {400,600},
5051 <bch|ptm|ugm> . = { ,700},
5052 <blg>      {,}= {300,500},
5053 <m-t|cmr|pad|pmn|ppl> {,}= { ,500},
5054 <bch|ugm>    {,}= { ,600},
5055 <ptm>      {,}= { ,700},
5056 <m-t|cmr|pad|ppl> : = { ,300},
5057 <bch|ugm>    : = { ,400},
5058 <pmn>      : = { ,200},
5059 <ptm>      : = { ,500},
5060 <m-t|cmr|pad|ppl> ; = { ,300},
5061 <bch|ugm>    ; = { ,400},
5062 <pmn>      ; = { ,200},
5063 <ptm>      ; = { ,500},
5064 <ptm>      ! = { ,100},
5065 <bch>      ? = { ,200},
5066 <ptm>      ? = { ,100},
5067 <ppl>      ? = { ,300},
5068 <pmn>      " = {400,200},
5069 <m-t|pad|pmn|ppl|ptm> & = {50,50},
5070 <bch>      & = { ,80},
5071 <cmr>      & = {100,50},
5072 <ugm>      & = {50,100},
5073 <m-t|cmr|pad|pmn> \% = {100, },
5074 <bch>      \% = {50,50},
5075 <ppl|ptm>    \% = {100,100},
5076 <ugm>      \% = {100,50},
5077 <m-t|pmn|ppl> * = {200,200},
5078 <bch>      * = {300,200},
5079 <cmr>      * = {400,100},
5080 <pad>      * = {500,100},
5081 <ptm|ugm>   * = {400,200},
5082 <m-t|cmr|pmn|ppl> + = {150,200},
5083 <bch|ugm>    + = {250,250},
5084 <pad|ptm>    + = {250,200},
5085 <m-t|pad|pmn|ppl> @ = {50,50},
5086 <bch>      @ = {80,50},
5087 <cmr>      @ = {200,50},
5088 <ptm>      @ = {150,150},
5089 <m-t|bch|ugm> ~ = {150,150},
5090 <cmr|pad|pmn|ppl|ptm> ~ = {200,150},
5091 <ugm>      {=} = {200,200},
5092 <!blg>      ( = {200, }, ) = { ,200},
5093 <m-t|cmr|pad|ppl|ptm|ugm> / = {100,200},
5094 <bch>      / = { ,150},
5095 <pmn>      / = {100,150},

```

```

5096 <m-t>      - = {300,300},
5097 <bch|pad>   - = {300,400},
5098 <pmn>       - = {200,300},
5099 <cmr>       - = {500,300},
5100 <ppl>       - = {300,500},
5101 <ptm>       - = {500,500},
5102 <ugm>       - = {400,700},
5103 <blg>       - = {0,300},
5104 <m-t|pmn>   \textendash      = {200,200}, \textemdash      = {150,150},
5105 <bch>       \textendash      = {200,300}, \textemdash      = {150,200},
5106 <cmr>       \textendash      = {500,300}, \textemdash      = {400,200},
5107 <pad|ppl|ptm|ugm> \textendash      = {300,300}, \textemdash      = {200,200},
5108 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
5109 <blg>       \textquoteleft = {400,400}, \textquoteright = {400,400},
5110 <cmr|pad>   \textquoteleft = {800,200}, \textquoteright = {800,200},
5111 <ppl>       \textquoteleft = {700,400}, \textquoteright = {700,400},
5112 <ptm>       \textquoteleft = {800,500}, \textquoteright = {800,500},
5113 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
5114 <blg>       \textquotedblright = {300,300}
5115 <cmr>       \textquotedblleft = {700,100}, \textquotedblright = {500,300}
5116 <pad>       \textquotedblleft = {700,200}, \textquotedblright = {700,200}
5117 <ppl>       \textquotedblleft = {500,300}, \textquotedblright = {500,300}
5118 <ptm>       \textquotedblleft = {700,400}, \textquotedblright = {700,400}
5119 <ugm>       \textquotedblleft = {600,200}, \textquotedblright = {600,200}
5120   }
5121
5122 <*cmr>
5123 \SetProtrusion
5124   [ name      = cmr-it-OT1,
5125     load      = cmr-it   ]
5126   { encoding = {OT1,OT4},
5127     family   = cmr,
5128     shape    = it       }
5129   {
5130     \AE = {100,   },
5131     \OE = {100,   },
5132     "00 = {200,150}, % \Gamma
5133     "01 = {150,100}, % \Delta
5134     "02 = {150, 50}, % \Theta
5135     "03 = {150, 50}, % \Lambda
5136     "04 = {100,100}, % \Xi
5137     "05 = {100,100}, % \Pi
5138     "06 = {100, 50}, % \Sigma
5139     "07 = {200,150}, % \Upsilon
5140     "08 = {150, 50}, % \Phi
5141     "09 = {150,100}, % \Psi
5142     "0A = { 50, 50} % \Omega
5143   }
5144
5145 </cmr>
5146 \SetProtrusion
5147 <m-t>   [ name      = T1-it-default,
5148 <bch>   [ name      = bch-it-T1,
5149 <blg>   [ name      = blg-it-T1,
5150 <cmr>   [ name      = cmr-it-T1,
5151 <pad>   [ name      = pad-it-T1,
5152 <pmn>   [ name      = pmnj-it-T1,
5153 <ppl>   [ name      = ppl-it-T1,
5154 <ptm>   [ name      = ptm-it-T1,
5155 <ugm>   [ name      = ugm-it-T1,
5156 <m-t>   load      = OT1-it   ]
5157 <bch>   load      = bch-it   ]
5158 <blg>   load      = blg-T1   ]

```

```

5159 <cmr>      load      = cmr-it   ]
5160 <pmn>      load      = pmnj-it  ]
5161 <pad>      load      = pad-it   ]
5162 <ppl>      load      = ppl-it   ]
5163 <ptm>      load      = ptm-it   ]
5164 <ugm>      load      = ugm-it   ]
5165 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
5166 <blg|ptm|ugm> { encoding = T1,
5167 <bch>      family    = bch,
5168 <blg>      family    = blg,
5169 <cmr>      family    = cmr,
5170 <pmn>      family    = pmnj,
5171 <pad>      family    = {pad,padx,padj},
5172 <ppl>      family    = {ppl,pplx,pplj},
5173 <ptm>      family    = {ptm,ptmx,ptmj},
5174 <ugm>      family    = ugm,
5175 <m-t|bch|pad|pmn|ppl|ptm> shape    = {it,sl} }
5176 <blg|cmr|ugm> shape    = it      }
5177 {
5178 <m-t|bch|pmn>      _ = { ,100},
5179 <blg>              _ = {0,300},
5180 <cmr|ugm>          _ = {100,200},
5181 <pad|ppl|ptm>      _ = {100,100},
5182 <blg>              . = {400,600},
5183 <blg>              {,}= {300,500},
5184 <cmr>              \AE = {100,  },
5185 <bch>              \OE = { 50,  },
5186 <cmr>              \OE = {100,  },
5187 <pmn>              031 = { , -100}, % ffl
5188 <cmr|ptm>          156 = {100, }, % IJ
5189 <pad>              156 = {50,  }, % IJ
5190 <pmn>              156 = {20,  }, % IJ
5191 <pmn>              188 = { , -30}, % ij
5192 <pmn>              \v t = { ,100},
5193 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
5194 <cmr|ugm>          \textbackslash = {300,300},
5195 <bch>              \textbackslash = {150,150},
5196 <pmn>              \textbackslash = {100,150},
5197 <ugm>              \textbar      = {200,200},
5198 <cmr>              \textquotedblleft = {500,300},
5199 <blg>              \textquoteleft  = {400,400}, \textquoteright = {400,400},
5200 <blg>              \textquotedbl = {300,300}, \textquotedblleft = {300,300},
5201 <blg>              \textquotedblright = {300,300}, \quotedblbase = {200,600},
5202 <m-t|ptm>          \quotesinglbase = {300,700}, \quotedblbase = {400,500},
5203 <cmr>              \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5204 <bch|pmn>          \quotesinglbase = {200,500}, \quotedblbase = {150,500},
5205 <pad|ppl>          \quotesinglbase = {500,500}, \quotedblbase = {400,400},
5206 <ugm>              \quotesinglbase = {300,700}, \quotedblbase = {300,500},
5207 <m-t|ppl|ptm>      \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5208 <bch|pmn>          \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5209 <cmr>              \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5210 <pad>              \guilsinglleft = {500,400}, \guilsinglright = {300,500},
5211 <ugm>              \guilsinglleft = {400,400}, \guilsinglright = {300,600},
5212 <m-t|ppl>          \guillemotleft = {300,300}, \guillemotright = {300,300},
5213 <bch|pmn>          \guillemotleft = {200,300}, \guillemotright = {150,400},
5214 <cmr>              \guillemotleft = {400,100}, \guillemotright = {200,300},
5215 <pad>              \guillemotleft = {300,300}, \guillemotright = {200,400},
5216 <ptm>              \guillemotleft = {300,400}, \guillemotright = {200,400},
5217 <ugm>              \guillemotleft = {300,400}, \guillemotright = {300,400},
5218 <m-t|pad|ppl|ugm> \textexclamdown = {100,  }, \textquestiondown = {200,  },
5219 <cmr|ptm>          \textexclamdown = {200,  }, \textquestiondown = {200,  },
5220 <pmn>              \textexclamdown = {-50,  }, \textquestiondown = {-50,  },
5221 <m-t|ppl|ugm>      \textbraceleft  = {200,100}, \textbraceright = {200,200},

```

```

5222 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
5223 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
5224 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
5225 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
5226 <pmn> \textvisiblespace = {100,100}
5227 }
5228
5229 <*m-t|cmr>
5230 \SetProtrusion
5231 <m-t> [ name = T2A-it-default,
5232 <cmr> [ name = cmr-it-T2A,
5233 <m-t> load = OT1-it ]
5234 <cmr> load = cmr-it ]
5235 { encoding = T2A,
5236 <m-t> shape = {it,sl} }
5237 <cmr> family = cmr,
5238 <cmr> shape = it }
5239 {
5240 <*cmr>
5241 \CYRA = {100,50},
5242 \CYRB = {50, },
5243 \CYRV = {50, },
5244 \CYRG = {100, },
5245 \CYRD = {50, },
5246 \CYRE = {50, },
5247 \CYRZH = {50, },
5248 \CYRZ = {50, },
5249 \CYRI = {50, },
5250 \CYRISHRT = {50, },
5251 \CYRK = {50, },
5252 \CYRL = {50, },
5253 \CYRM = {50, },
5254 \CYRN = {50, },
5255 \CYRO = {100, },
5256 \CYRP = {50, },
5257 \CYRR = {50, },
5258 \CYRS = {100, },
5259 \CYRT = {100, },
5260 \CYRU = {100, },
5261 \CYRF = {100, },
5262 \CYRH = {50, },
5263 \CYRC = {50, },
5264 \CYRCH = {100, },
5265 \CYRSH = {50, },
5266 \CYRSHCH = {50, },
5267 \CYRHRDSN = {100, },
5268 \CYRERY = {50, },
5269 \CYRSFTSN = {50, },
5270 \CYREREV = {50, },
5271 \CYRYU = {50, },
5272 \CYRYA = {50, },
5273 _ = {100,200},
5274 </cmr>
5275 <m-t> _ = { ,100},
5276 <m-t> \textbackslash = {100,200}, \quotedblbase = {400,500},
5277 <cmr> \textbackslash = {300,300}, \quotedblbase = {200,600},
5278 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
5279 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
5280 <m-t> \textbraceleft = {200,100}, \textbraceright = {200,200},
5281 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
5282 <cmr> \textquotedblleft = {500,300},
5283 <cmr> \textless = {300,100}, \textgreater = {200,100}
5284 }

```



```

5285
5286 </m-t|cmr>
5287 <*m-t|ptm>
5288 \SetProtrusion
5289 <m-t> [ name = QX-it-default,
5290 <ptm> [ name = ptm-it-QX,
5291 <m-t> load = OT1-it ]
5292 <ptm> load = ptm-it ]
5293 { encoding = {QX},
5294 <ptm> family = {ptm,ptmx,ptmj},
5295 shape = {it,sl} }
5296 {
5297 <ptm> 009 = { , 50}, % fk
5298 {=} = {100,100},
5299 <m-t> \textunderscore = {100,100},
5300 <ptm> \textunderscore = {100,150},
5301 \textbackslash = {100,200},
5302 \quotedblbase = {300,400},
5303 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
5304 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
5305 \textexclamdown = {200, }, \textquestiondown = {200, },
5306 \textbraceleft = {200,100}, \textbraceright = {200,200},
5307 \textless = {100,100}, \textgreater = {100,100},
5308 \textminus = {200,200}, \textdegree = {300,150},
5309 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5310 <ptm> \textregistered = {100,150}, \copyright = {100,150},
5311 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
5312 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
5313 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
5314 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
5315 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
5316 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
5317 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5318 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
5319 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
5320 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
5321 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5322 <ptm> \textperthousand = { ,50}
5323 }
5324
5325 </m-t|ptm>
5326 <*cmr|bch>
5327 \SetProtrusion
5328 <cmr> [ name = cmr-it-T5,
5329 <cmr> load = cmr-it ]
5330 <bch> [ name = bch-it-T5,
5331 <bch> load = bch-it ]
5332 { encoding = T5,
5333 <bch> family = bch,
5334 <cmr> family = cmr,
5335 shape = it }
5336 {
5337 <bch> _ = { ,100},
5338 <cmr> _ = {100,200},
5339 <bch> \textbackslash = {150,150},
5340 <cmr> \textbackslash = {300,300},
5341 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
5342 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5343 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5344 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5345 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
5346 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
5347 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},

```

```

5348 <cmr>      \textbraceleft   = {400,100}, \textbraceright   = {200,200},
5349 <bch>      \textless         = {100,  }, \textgreater      = {  ,100}
5350 <cmr>      \textless         = {300,100}, \textgreater      = {200,100}
5351   }
5352
5353 </cmr|bch>

```

Slanted is very similar to italic.

```

5354 <*cmr>
5355 \SetProtrusion
5356   [ name      = cmr-sl,
5357     load      = cmr-it-OT1 ]
5358   { encoding = {OT1,OT4},
5359     family   = cmr,
5360     shape    = sl  }
5361   {
5362     L = {  ,50},
5363     f = {  ,-50},
5364     - = {300, },
5365     \textendash = {400,  }, \textemdash = {300,  }
5366   }
5367
5368 \SetProtrusion
5369   [ name      = cmr-sl-T1,
5370     load      = cmr-it-T1 ]
5371   { encoding = {T1,LY1},
5372     family   = cmr,
5373     shape    = sl  }
5374   {
5375     L = {  ,50},
5376     f = {  ,-50},
5377     - = {300, },
5378     \textendash = {400,  }, \textemdash = {300,  }
5379   }
5380
5381 \SetProtrusion
5382   [ name      = cmr-sl-T2A,
5383     load      = cmr-it-T2A ]
5384   { encoding = T2A,
5385     family   = cmr,
5386     shape    = sl  }
5387   {
5388     L = {  ,50},
5389     f = {  ,-50},
5390     - = {300, },
5391     \textendash = {400,  }, \textemdash = {300,  }
5392   }
5393
5394 \SetProtrusion
5395   [ name      = cmr-sl-T5,
5396     load      = cmr-it-T5 ]
5397   { encoding = T5,
5398     family   = cmr,
5399     shape    = sl  }
5400   {
5401     L = {  ,50},
5402     f = {  ,-50},
5403     - = {300, },
5404     \textendash = {400,  }, \textemdash = {300,  }
5405   }
5406
5407 \SetProtrusion
5408   [ name      = lmr-it-T1,

```

```

5409     load      = cmr-it-T1 ]
5410   { encoding = {T1,LY1},
5411     family   = lmr,
5412     shape    = {it,sl} }
5413   {
5414     \textquotedblleft = { ,200}, \textquotedblright = { ,200},
5415     \quotesinglbase   = { ,400}, \quotedblbase     = { ,500}
5416   }
5417

```

Oldstyle numerals are slightly different.

```

5418 \SetProtrusion
5419 [ name = cmr(oldstyle)-it,
5420   load = cmr-it-T1 ]
5421 { encoding = T1,
5422   family   = {hfor,cmr},
5423   shape    = {it,sl} }
5424 {
5425   1 = {250, 50},
5426   2 = {150,-100},
5427   3 = {100,-50},
5428   4 = {150,150},
5429   6 = {200,  },
5430   7 = {200, 50},
5431   8 = {150,-50},
5432   9 = {100, 50}
5433 }
5434
5435 </cmr>
5436 <*pmn>
5437 \SetProtrusion
5438 [ name   = pmnx-it,
5439   load   = pmnj-it ]
5440 { encoding = OT1,
5441   family   = pmnx,
5442   shape    = {it,sl} }
5443 {
5444   1 = {100,150}
5445 }
5446
5447 \SetProtrusion
5448 [ name   = pmnx-it-T1,
5449   load   = pmnj-it-T1 ]
5450 { encoding = {T1,LY1},
5451   family   = pmnx,
5452   shape    = {it,sl} }
5453 {
5454   1 = {100,150}
5455 }
5456
5457 </pmn>
5458 <*ptm>
5459 \SetProtrusion
5460 [ name   = ptm-it-LY1,
5461   load   = ptm-it-T1 ]
5462 { encoding = {LY1},
5463   family   = {ptm,ptmx,ptmj},
5464   shape    = {it,sl} }
5465 {
5466   - = {100,100},
5467   \texttrademark = {100,100},
5468   \textregistered = {100,100},
5469   \textcopyright = {100,100},

```

```

5470 \textdegree = {300,100},
5471 \textminus = {200,200},
5472 \textellipsis = {100,200},
5473 % \texteuro = { , }, % ?
5474 \textcent = {100,100},
5475 \textquotesingle = {500, },
5476 \textflorin = {100, 70},
5477 \textdagger = {150,150},
5478 \textdaggerdbl = {100,100},
5479 \textbullet = {150,150},
5480 \textonesuperior = {150,100},
5481 \texttwosuperior = {150, 50},
5482 \textthreesuperior = {150, 50},
5483 \textparagraph = {100, },
5484 \textperiodcentered = {500,300},
5485 \textonequarter = { 50, },
5486 \textonehalf = { 50, },
5487 \textplusminus = {100,100},
5488 \textmultiply = {150,150},
5489 \textdivide = {150,150}
5490 }
5491
5492 </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).

```

5493 <*(big|ugm)>
5494 \SetProtrusion
5495 <m-t> [ name = OT1-sc,
5496 <bch> [ name = bch-sc,
5497 <cmr> [ name = cmr-sc-OT1,
5498 <pad> [ name = pad-sc,
5499 <pmn> [ name = pmnj-sc,
5500 <ppl> [ name = ppl-sc,
5501 <ptm> [ name = ptm-sc,
5502 <m-t> load = default ]
5503 <bch> load = bch-default ]
5504 <cmr> load = cmr-OT1 ]
5505 <pad> load = pad-default ]
5506 <pmn> load = pmnj-default ]
5507 <ppl> load = ppl-default ]
5508 <ptm> load = ptm-default ]
5509 <m-t|bch|pad|pmn> { encoding = OT1,
5510 <cmr|ppl|ptm> { encoding = {OT1,OT4},
5511 <bch> family = bch,
5512 <cmr> family = cmr,
5513 <pad> family = {pad,padx,padj},
5514 <pmn> family = pmnj,
5515 <ppl> family = {ppl,pplx,pplj},
5516 <ptm> family = {ptm,ptmx,ptmj},
5517 shape = sc }
5518 {
5519 a = {50,50},
5520 <cmr|pad|ppl|ptm> \ae = {50, },
5521 <bch|pmn> c = {50, },
5522 <bch|pad|pmn> d = { ,50},
5523 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
5524 <bch|pad|pmn> g = {50, },

```

```

5525 <m-t|cmr|pad|pmn|ppl|ptm>    j = {50,  },
5526 <bch>    j = {100,  },
5527 <m-t|bch|cmr|pad|pmn|ppl>    l = {  ,50},
5528 <ptm>    l = {  ,80},
5529 <m-t|bch|cmr|pad|pmn|ppl>    013 = {  ,50}, % f1
5530 <ptm>    013 = {  ,80}, % f1
5531 <bch|pad|pmn>    o = {50,50},
5532 <pad|pmn>    \oe = {50,  },
5533 <ppl>    p = { 0, 0},
5534 <bch|pad|pmn>    q = {50,70},
5535 <ppl>    q = { 0,  },
5536 <m-t|cmr|pad|pmn|ppl|ptm>    r = {  , 0},
5537    t = {50,50},
5538 <m-t|bch|cmr|pad|pmn|ppl>    y = {50,50}
5539 <ptm>    y = {80,80}
5540 }
5541
5542 \SetProtrusion
5543 <m-t>    [ name      = T1-sc,
5544 <bch>    [ name      = bch-sc-T1,
5545 <cmr>    [ name      = cmr-sc-T1,
5546 <pad>    [ name      = pad-sc-T1,
5547 <pmn>    [ name      = pmnj-sc-T1,
5548 <ppl>    [ name      = ppl-sc-T1,
5549 <ptm>    [ name      = ptm-sc-T1,
5550 <m-t>    load      = T1-default ]
5551 <bch>    load      = bch-T1    ]
5552 <cmr>    load      = cmr-T1    ]
5553 <pad>    load      = pad-T1    ]
5554 <pmn>    load      = pmnj-T1   ]
5555 <ppl>    load      = ppl-T1    ]
5556 <ptm>    load      = ptm-T1   ]
5557 { encoding = {T1,LY1},
5558 <bch>    family    = bch,
5559 <cmr>    family    = cmr,
5560 <pad>    family    = {pad,padx,padj},
5561 <pmn>    family    = pmnj,
5562 <ppl>    family    = {ppl,pplx,pplj},
5563 <ptm>    family    = {ptm,ptmx,ptmj},
5564    shape      = sc }
5565 {
5566    a = {50,50},
5567 <cmr|pad|ppl|ptm>    \ae = {50,  },
5568 <bch|pmn>    c = {50,  },
5569 <bch|pad|pmn>    d = {  ,50},
5570 <m-t|bch|cmr|pad|pmn|ptm>    f = {  ,50},
5571 <bch|pad|pmn>    g = {50,  },
5572 <m-t|cmr|pad|pmn|ppl|ptm>    j = {50,  },
5573 <bch>    j = {100,  },
5574 <m-t|bch|cmr|pad|pmn|ppl>    l = {  ,50},
5575 <ptm>    l = {  ,80},
5576 <m-t|bch|cmr|pad|pmn|ppl>    029 = {  ,50}, % f1
5577 <ptm>    029 = {  ,80}, % f1
5578 <bch|pad|pmn>    o = {50,50},
5579 <bch|pad|pmn>    \oe = {50,  },
5580 <ppl>    p = { 0, 0},
5581 <bch|pad|pmn>    q = {50,70},
5582 <ppl>    q = { 0,  },
5583 <m-t|cmr|pad|pmn|ppl|ptm>    r = {  , 0},
5584    t = {50,50},
5585 <m-t|bch|cmr|pad|pmn|ppl>    y = {50,50}
5586 <ptm>    y = {80,80}
5587 }
```

```

5588
5589 </!(blg|ugm)>
5590 <*m-t|cmr>
5591 \SetProtrusion
5592 <m-t> [ name      = T2A-sc,
5593 <cmr> [ name      = cmr-sc-T2A,
5594 <m-t>   load      = T2A-default ]
5595 <cmr>   load      = cmr-T2A   ]
5596 { encoding = T2A,
5597 <cmr>   family   = cmr,
5598   shape   = sc }
5599 {
5600   \cyra = {50,50},
5601   \cyrg = { ,50},
5602   \cyrt = {50,50},
5603   \cyrj = { ,50}
5604 }
5605
5606 </m-t|cmr>
5607 <*m-t>
5608 \SetProtrusion
5609 [ name      = QX-sc,
5610   load      = QX-default ]
5611 { encoding = QX,
5612   shape     = sc }
5613 {
5614   a = {50,50},
5615   f = { ,50},
5616   j = {50, },
5617   l = { ,50},
5618   013 = { ,50}, % fl
5619   r = { , 0},
5620   t = {50,50},
5621   y = {50,50}
5622 }
5623
5624 </m-t>
5625 <*cmr|bch>
5626 \SetProtrusion
5627 <bch> [ name      = bch-sc-T5,
5628 <bch>   load      = bch-T5 ]
5629 <cmr> [ name      = cmr-sc-T5,
5630 <cmr>   load      = cmr-T5 ]
5631 { encoding = T5,
5632 <bch>   family   = bch,
5633 <cmr>   family   = cmr,
5634   shape   = sc }
5635 {
5636   a = {50,50},
5637 <bch>   c = {50, },
5638 <bch>   d = { ,50},
5639   f = { ,50},
5640 <bch>   g = {50, },
5641 <bch>   j = {100, },
5642 <cmr>   j = {50, },
5643   l = { ,50},
5644 <bch>   o = {50,50},
5645 <bch>   q = { 0, },
5646 <cmr>   r = { , 0},
5647   t = {50,50},
5648   y = {50,50}
5649 }
5650

```

```

5651 </cmr|bch>
5652 < *pmn>
5653 \SetProtrusion
5654   [ name      = pmnx-sc,
5655     load      = pmnj-sc ]
5656   { encoding = OT1,
5657     family   = pmnx,
5658     shape    = sc }
5659   {
5660     1 = {230,180}
5661   }
5662
5663 \SetProtrusion
5664   [ name      = pmnx-sc-T1,
5665     load      = pmnj-sc-T1 ]
5666   { encoding = {T1,LY1},
5667     family   = pmnx,
5668     shape    = sc }
5669   {
5670     1 = {230,180}
5671   }
5672

```

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`.

```

5673 \SetProtrusion
5674   [ name      = pmnj-scit,
5675     load      = pmnj-it ]
5676   { encoding = OT1,
5677     family   = pmnj,
5678     shape    = {scit,si} }
5679   {
5680     a = {50, },
5681     \ae = { , -50},
5682     b = {20, -50},
5683     c = {50, -50},
5684     d = {20, 0},
5685     e = {20, -50},
5686     f = {10, 0},
5687     012 = {10, -50}, % fi
5688     013 = {10, -50}, % fl
5689     014 = {10, -50}, % ffi
5690     015 = {10, -50}, % ffl
5691     g = {50, -50},
5692     i = {20, -50},
5693     j = {20, 0},
5694     k = {20, },
5695     l = {20, 50},
5696     m = { , -30},
5697     n = { , -30},
5698     o = {50, },
5699     \oe = {50, -50},
5700     p = {20, -50},
5701     q = {50, },
5702     r = {20, 0},
5703     s = {20, -30},
5704     t = {70, },
5705     u = {50, -50},
5706     v = {100, },

```

```

5707     w = {100, },
5708     y = {50, },
5709     z = { , -50}
5710 }
5711
5712 \SetProtrusion
5713 [ name      = pmnj-scit-T1,
5714   load      = pmnj-it-T1 ]
5715 { encoding = {T1,LY1},
5716   family   = pmnj,
5717   shape     = {scit,si} }
5718 {
5719   a = {50, },
5720   \ae = { , -50},
5721   b = {20, -50},
5722   c = {50, -50},
5723   d = {20, 0},
5724   e = {20, -50},
5725   f = {10, 0},
5726   028 = {10, -50}, % fi
5727   029 = {10, -50}, % fl
5728   030 = {10, -50}, % ffi
5729   031 = {10, -50}, % ffl
5730   g = {50, -50},
5731   i = {20, -50},
5732   188 = {20, 0}, % ij
5733   j = {20, 0},
5734   k = {20, },
5735   l = {20, 50},
5736   m = { , -30},
5737   n = { , -30},
5738   o = {50, },
5739   \oe = {50, -50},
5740   p = {20, -50},
5741   q = {50, },
5742   r = {20, 0},
5743   s = {20, -30},
5744   t = {70, },
5745   u = {50, -50},
5746   v = {100, },
5747   w = {100, },
5748   y = {50, },
5749   z = { , -50}
5750 }
5751
5752 \SetProtrusion
5753 [ name      = pmnx-scit,
5754   load      = pmnj-scit ]
5755 { encoding = OT1,
5756   family   = pmnx,
5757   shape     = {scit,si} }
5758 {
5759   1 = {100,150}
5760 }
5761
5762 \SetProtrusion
5763 [ name      = pmnx-scit-T1,
5764   load      = pmnj-scit-T1 ]
5765 { encoding = {T1,LY1},
5766   family   = pmnx,
5767   shape     = {scit,si} }
5768 {
5769   1 = {100,150}

```



```

5770 }
5771
5772 (/pmn)

```

15.8.5 Text companion

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

```

5773 \SetProtrusion
5774 <m-t> [ name = textcomp ]
5775 <bch> [ name = bch-textcomp ]
5776 <blg> [ name = blg-textcomp ]
5777 <cmr> [ name = cmr-textcomp ]
5778 <pad> [ name = pad-textcomp ]
5779 <pmn> [ name = pmn-textcomp ]
5780 <ppl> [ name = ppl-textcomp ]
5781 <ptm> [ name = ptm-textcomp ]
5782 <ugm> [ name = ugm-textcomp ]
5783 <m-t> { encoding = TS1 }
5784 <!m-t> { encoding = TS1,
5785 <bch> family = bch }
5786 <blg> family = blg }
5787 <cmr> family = cmr }
5788 <pad> family = {pad,padx,padj} }
5789 <pmn> family = {pmnx,pmnj} }
5790 <ppl> family = {ppl,pplx,pplj} }
5791 <ptm> family = {ptm,ptmx,ptmj} }
5792 <ugm> family = ugm }
5793 {
5794 <blg> \textquotestraightbase = {400,500},
5795 <cmr> \textquotestraightbase = {300,300},
5796 <pad|pmn> \textquotestraightbase = {400,400},
5797 <blg> \textquotestraightdblbase = {300,400},
5798 <cmr|pmn> \textquotestraightdblbase = {300,300},
5799 <pad> \textquotestraightdblbase = {400,400},
5800 <bch|cmr|pad|pmn|ugm> \texttwelveudash = {200,200},
5801 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
5802 <ugm> \textthreequartersemdash = {200,200},
5803 <blg> \textquotesingle = {500,600},
5804 <cmr|pmn> \textquotesingle = {300,400},
5805 <pad> \textquotesingle = {400,500},
5806 <ptm> \textquotesingle = {500,500},
5807 <ugm> \textquotesingle = {300,500},
5808 <bch|cmr|pmn> \textasteriskcentered = {200,300},
5809 <blg> \textasteriskcentered = {150,200},
5810 <pad> \textasteriskcentered = {300,300},
5811 <ugm> \textasteriskcentered = {100,200},
5812 <pmn> \textfractionsolidus = {-200,-200},
5813 <cmr> \textoneoldstyle = {100,100},
5814 <pmn> \textoneoldstyle = { , 50},
5815 <cmr> \textthreeoldstyle = { , 50},
5816 <pad|pmn> \textthreeoldstyle = { 50, },
5817 <cmr> \textfouroldstyle = { 50, 50},
5818 <pad|pmn> \textfouroldstyle = { 50, },
5819 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
5820 <cmr> \textlangle = {400, },
5821 <cmr> \textrangle = { ,400},
5822 <m-t|bch|pmn|ptm> \textminus = {200,200},
5823 <cmr|pad|ppl> \textminus = {300,300},
5824 <blg|ugm> \textminus = {250,300},
5825 <bch|pad|pmn> \textlbrackdbl = {100, },

```

```

5826 <blg> \textlbrackdbl = {200, },
5827 <bch|pad|pmn> \textrbrackdbl = { ,100},
5828 <blg> \textrbrackdbl = { ,200},
5829 <pmn> \textasciigrave = {200,500},
5830 <bch|blg|cmr|pad|pmn> \texttildebelow = {200,250},
5831 <pmn> \textasciibreve = {300,400},
5832 <pmn> \textasciicaron = {300,400},
5833 <pmn> \textacutedbl = {200,300},
5834 <pmn> \textgravedbl = {150,300},
5835 <bch|pmn|ugm> \textdagger = { 80, 80},
5836 <blg> \textdagger = {200,200},
5837 <cmr|pad> \textdagger = {100,100},
5838 <ptm> \textdagger = {150,150},
5839 <blg> \textdaggerdbl = {150,150},
5840 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
5841 <ptm> \textdaggerdbl = {100,100},
5842 <bch> \textbardbl = {100,100},
5843 <blg|ugm> \textbardbl = {150,150},
5844 <bch> \textbullet = {200,200},
5845 <blg> \textbullet = {400,500},
5846 <cmr|pad|pmn> \textbullet = { ,100},
5847 <ptm> \textbullet = {150,150},
5848 <ugm> \textbullet = { 50,100},
5849 <bch|cmr|pmn> \textcelsius = { 50, },
5850 <pad> \textcelsius = { 80, },
5851 <bch> \textflorin = { 50, 50},
5852 <blg> \textflorin = {100,100},
5853 <pad|ugm> \textflorin = { ,100},
5854 <pmn> \textflorin = { 50,100},
5855 <ptm> \textflorin = { 50, 70},
5856 <cmr> \textcolonmonetary = { , 50},
5857 <pad|pmn> \textcolonmonetary = { 50, },
5858 <pmn> \textinterrobang = { ,100},
5859 <pmn> \textinterrobangdown = {100, },
5860 <m-t|pad|ptm> \texttrademark = {100,100},
5861 <bch> \texttrademark = {150,150},
5862 <blg|cmr|ppl> \texttrademark = {200,200},
5863 <pmn> \texttrademark = { 50, 50},
5864 <ugm> \texttrademark = {100,150},
5865 <bch|ugm> \textcent = { 50, },
5866 <ptm> \textcent = {100,100},
5867 <bch> \textsterling = { 50, },
5868 <ugm> \textsterling = { , 50},
5869 <bch> \textbrokenbar = {200,200},
5870 <blg> \textbrokenbar = {250,250},
5871 <ugm> \textbrokenbar = {200,300},
5872 <pmn> \textasciidieresis = {300,400},
5873 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
5874 <pmn> \textcopyright = {100,150},
5875 <ppl> \textcopyright = {200,200},
5876 <bch|cmr|ugm> \textordfeminine = {100,200},
5877 <pad|pmn> \textordfeminine = {200,200},
5878 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
5879 <blg> \textlnot = {200,100},
5880 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
5881 <pmn> \textregistered = { 50,150},
5882 <ppl> \textregistered = {200,200},
5883 <pmn> \textasciimacron = {150,200},
5884 <m-t|ppl|ptm> \textdegree = {300,300},
5885 <bch> \textdegree = {150,200},
5886 <blg|ugm> \textdegree = {200,200},
5887 <cmr|pad> \textdegree = {400,400},
5888 <pmn> \textdegree = {150,400},

```

```

5889 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
5890 <blg> \textpm = {100,100},
5891 <ptm> \textpm = { 50, 80},
5892 <bch|blg|ugm> \texttwosuperior = {100,200},
5893 <cmr> \texttwosuperior = { 50,100},
5894 <pad|pmn> \texttwosuperior = {200,200},
5895 <ptm> \texttwosuperior = { 50, 50},
5896 <bch|blg|ugm> \textthreesuperior = {100,200},
5897 <cmr> \textthreesuperior = { 50,100},
5898 <pad|pmn> \textthreesuperior = {200,200},
5899 <ptm> \textthreesuperior = { 50, 50},
5900 <pmn> \textasciicute = {300,400},
5901 <bch|ugm> \textmu = { ,100},
5902 <bch|pad|pmn> \textparagraph = { ,100},
5903 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
5904 <blg> \textperiodcentered = {400,500},
5905 <ptm> \textperiodcentered = {300,300},
5906 <ugm> \textperiodcentered = {200,500},
5907 <bch|blg|ugm> \textonesuperior = {200,300},
5908 <cmr|pad|pmn> \textonesuperior = {200,200},
5909 <ptm> \textonesuperior = {100,100},
5910 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
5911 <blg|cmr> \textordmasculine = {100,200},
5912 <bch|cmr|pmn> \texteuro = {100, },
5913 <pad> \texteuro = { 50,100},
5914 <bch> \texttimes = {200,200},
5915 <blg|ptm> \texttimes = {100,100},
5916 <cmr> \texttimes = {150,250},
5917 <pad> \texttimes = {100,150},
5918 <pmn> \texttimes = { 70,100},
5919 <ugm> \texttimes = {200,300},
5920 <bch|pad|pmn> \textdiv = {150,200}
5921 <blg> \textdiv = {100,100}
5922 <cmr> \textdiv = {150,250}
5923 <ptm> \textdiv = { 50,100},
5924 <ugm> \textdiv = {200,300},
5925 <ptm> \textperthousand = { ,50},
5926 <ugm> \textsection = { ,100},
5927 <ugm> \textonehalf = { 50,100},
5928 <ugm> \textonequarter = { 50,100},
5929 <ugm> \textthreequarters = { 50,100},
5930 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

5931 }
5932
5933 <*cmr|pad|pmn|ugm>
5934 \SetProtrusion
5935 <cmr> [ name = cmr-textcomp-it ]
5936 <pad> [ name = pad-textcomp-it ]
5937 <pmn> [ name = pmn-textcomp-it ]
5938 <ugm> [ name = ugm-textcomp-it ]
5939 { encoding = TS1,
5940 <cmr> family = cmr,
5941 <pad> family = {pad,padx,padj},
5942 <pmn> family = {pmnx,pmnj},
5943 <ugm> family = ugm,
5944 <!ugm> shape = {it,sl} }
5945 <ugm> shape = it }
5946 {
5947 <cmr> \textquotestraightbase = {300,600},
5948 <pad|pmn> \textquotestraightbase = {400,400},
5949 <cmr> \textquotestraightdblbase = {300,600},

```

```

5950 <pad> \textquotestraightdblbase = {300,400},
5951 <pmn> \textquotestraightdblbase = {300,300},
5952 \texttwelvewardash = {200,200},
5953 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
5954 <ugm> \textthreequartersemdash = {200,200},
5955 <cmr> \textquotesingle = {600,300},
5956 <pad> \textquotesingle = {800,100},
5957 <pmn> \textquotesingle = {300,200},
5958 <ugm> \textquotesingle = {500,500},
5959 <cmr> \textasteriskcentered = {300,200},
5960 <pad> \textasteriskcentered = {500,100},
5961 <pmn> \textasteriskcentered = {200,300},
5962 <ugm> \textasteriskcentered = {300,150},
5963 <pmn> \textfractionsolidus = {-200,-200},
5964 <cmr> \textoneoldstyle = {100, 50},
5965 <pad> \textoneoldstyle = {100, },
5966 <pmn> \textoneoldstyle = { 50, },
5967 <pad> \texttwooldstyle = { 50, },
5968 <pmn> \texttwooldstyle = {-50, },
5969 <cmr> \textthreeoldstyle = {100, 50},
5970 <pmn> \textthreeoldstyle = {-100, },
5971 <cmr> \textfouroldstyle = { 50, 50},
5972 <pad> \textfouroldstyle = { 50,100},
5973 <cmr> \textsevenoldstyle = { 50, 80},
5974 <pad> \textsevenoldstyle = { 50, },
5975 <pmn> \textsevenoldstyle = { 20, },
5976 <cmr> \textlangle = {400, },
5977 <cmr> \textrangle = { ,400},
5978 <cmr|pad> \textminus = {300,300},
5979 <pmn> \textminus = {200,200},
5980 <ugm> \textminus = {250,300},
5981 <pad|pmn> \textlbrackdbl = {100, },
5982 <pad|pmn> \textrbrackdbl = { ,100},
5983 <pmn> \textasciigrave = {300,300},
5984 <cmr|pad|pmn> \texttildelow = {200,250},
5985 <pmn> \textasciibreve = {300,300},
5986 <pmn> \textasciicaron = {300,300},
5987 <pmn> \textacutedbl = {200,300},
5988 <pmn> \textgravedbl = {150,300},
5989 <cmr> \textdagger = {100,100},
5990 <pad> \textdagger = {200,100},
5991 <pmn> \textdagger = { 80, 50},
5992 <ugm> \textdagger = { 80, 80},
5993 <cmr|pad> \textdaggerdbl = { 80, 80},
5994 <pmn> \textdaggerdbl = { 80, 50},
5995 <ugm> \textbardbl = {150,150},
5996 <cmr> \textbullet = {200,100},
5997 <pad> \textbullet = {300, },
5998 <pmn> \textbullet = { 30, 70},
5999 <ugm> \textbullet = { 50,100},
6000 <cmr> \textcelsius = {100, },
6001 <pad> \textcelsius = {200, },
6002 <pmn> \textcelsius = { 50,-50},
6003 <pad> \textflorin = {100, },
6004 <pmn> \textflorin = { 50,100},
6005 <ugm> \textflorin = { ,100},
6006 <cmr> \textcolonmonetary = {150, },
6007 <pad> \textcolonmonetary = {100, },
6008 <pmn> \textcolonmonetary = { 50,-50},
6009 <cmr|pad> \texttrademark = {200, },
6010 <pmn> \texttrademark = { 50,100},
6011 <ugm> \texttrademark = {150, 50},
6012 <ugm> \textcent = { 50, },

```

```

6013 <ugm> \textsterling = { , 50},
6014 <ugm> \textbrokenbar = {200,300},
6015 <pmn> \textasciidieresis = {300,200},
6016 <cmr> \textcopyright = {100, },
6017 <pad> \textcopyright = {200,100},
6018 <pmn> \textcopyright = {100,150},
6019 <ugm> \textcopyright = {300, },
6020 <cmr> \textordfeminine = {100,100},
6021 <pmn> \textordfeminine = {200,200},
6022 <ugm> \textordfeminine = {100,200},
6023 <cmr|pad> \textlnot = {300, },
6024 <pmn|ugm> \textlnot = {200, },
6025 <cmr> \textregistered = {100, },
6026 <pad> \textregistered = {200,100},
6027 <pmn> \textregistered = { 50,150},
6028 <ugm> \textregistered = {300, },
6029 <pmn> \textasciimacron = {150,200},
6030 <cmr|pad> \textdegree = {500,100},
6031 <pmn> \textdegree = {150,150},
6032 <ugm> \textdegree = {300,200},
6033 <cmr> \textpm = {150,100},
6034 <pad> \textpm = {200,150},
6035 <pmn|ugm> \textpm = {150,200},
6036 <cmr> \textonesuperior = {400, },
6037 <pad> \textonesuperior = {300,100},
6038 <pmn> \textonesuperior = {200,100},
6039 <ugm> \textonesuperior = {300,300},
6040 <cmr> \texttwosuperior = {400, },
6041 <pad> \texttwosuperior = {300, },
6042 <pmn> \texttwosuperior = {200,100},
6043 <ugm> \texttwosuperior = {300,200},
6044 <cmr> \textthreesuperior = {400, },
6045 <pad> \textthreesuperior = {300, },
6046 <pmn> \textthreesuperior = {200,100},
6047 <ugm> \textthreesuperior = {300,200},
6048 <ugm> \textmu = { ,100},
6049 <pmn> \textasciicute = {300,200},
6050 <cmr> \textparagraph = {200, },
6051 <pmn> \textparagraph = { ,100},
6052 <cmr> \textperiodcentered = {500,500},
6053 <pad|pmn|ugm> \textperiodcentered = {300,400},
6054 <cmr> \textordmasculine = {100,100},
6055 <pmn> \textordmasculine = {200,200},
6056 <ugm> \textordmasculine = {300,200},
6057 <cmr> \texteuro = {200, },
6058 <pad> \texteuro = {100, },
6059 <pmn> \texteuro = {100,-50},
6060 <cmr> \texttimes = {200,200},
6061 <pad> \texttimes = {200,100},
6062 <pmn> \texttimes = { 70,100},
6063 <ugm> \texttimes = {200,300},
6064 <cmr|pad> \textdiv = {200,200},
6065 <pmn> \textdiv = {150,200},
6066 <ugm> \textdiv = {200,300},
6067 <ugm> \textsection = { ,200},
6068 <ugm> \textonehalf = { 50,100},
6069 <ugm> \textonequarter = { 50,100},
6070 <ugm> \textthreequarters = { 50,100},
6071 <ugm> \textsurd = { ,100}
6072 }
6073
6074 </cmr|pad|pmn|ugm>

```

15.8.6 Computer Modern math

Now to the math symbols for Computer Modern Roman. Definitions have been extracted from `fontmath.ltx`. I did not spend too much time fiddling with these settings, so they can surely be improved.

The math font ‘operators’ (also used for the `\mathrm` and `\mathbf` alphabets) is OT1/cmr, which we’ve already set up above. It’s declared as:

```
\DeclareSymbolFont{operators} {OT1}{cmr}{m}{n}
\SetSymbolFont{operators}{bold}{OT1}{cmr}{bx}{n}
```

`\mathit` (OT1/cmr/m/it) is also already set up.

There are (for the moment) no settings for `\mathsf` and `\mathtt`.

Math font ‘letters’ (also used as `\mathnormal`) is declared as:

```
\DeclareSymbolFont{letters} {OML}{cmm}{m}{it}
\SetSymbolFont{letters} {bold}{OML}{cmm}{b}{it}
```

```
6075 (*cmr*)
6076 \SetProtrusion
6077 [ name = cmr-math-letters ]
6078 { encoding = OML,
6079   family = cmm,
6080   series = {m,b},
6081   shape = it }
6082 {
6083   A = {100, 50}, % \mathnormal
6084   B = { 50,   },
6085   C = { 50,   },
6086   D = { 50, 50},
6087   E = { 50,   },
6088   F = {100, 50},
6089   G = { 50, 50},
6090   H = { 50, 50},
6091   I = { 50, 50},
6092   J = {150, 50},
6093   K = { 50,100},
6094   L = { 50, 50},
6095   M = { 50,   },
6096   N = { 50,   },
6097   O = { 50,   },
6098   P = { 50,   },
6099   Q = { 50, 50},
6100   R = { 50,   },
6101   S = { 50,   },
6102   T = { 50,100},
6103   U = { 50, 50},
6104   V = {100,100},
6105   W = { 50,100},
6106   X = { 50,100},
6107   Y = {100,100},
6108   f = {100,100},
6109   h = {   ,100},
6110   i = {   , 50},
6111   j = {   , 50},
6112   k = {   , 50},
6113   r = {   , 50},
6114   v = {   , 50},
6115   w = {   , 50},
6116   x = {   , 50},
6117   "OB = { 50,100}, % \alpha
```

```

6118 "0C = { 50, 50}, % \beta
6119 "0D = {200,150}, % \gamma
6120 "0E = { 50, 50}, % \delta
6121 "0F = { 50, 50}, % \epsilon
6122 "10 = { 50,150}, % \zeta
6123 "12 = { 50, }, % \theta
6124 "13 = { ,100}, % \iota
6125 "14 = { ,100}, % \kappa
6126 "15 = {100, 50}, % \lambda
6127 "16 = { , 50}, % \mu
6128 "17 = { , 50}, % \nu
6129 "18 = { , 50}, % \xi
6130 "19 = { 50,100}, % \pi
6131 "1A = { 50, 50}, % \rho
6132 "1B = { ,150}, % \sigma
6133 "1C = { 50,150}, % \tau
6134 "1D = { 50, 50}, % \upsilon
6135 "1F = { 50,100}, % \chi
6136 "20 = { 50, 50}, % \psi
6137 "21 = { , 50}, % \omega
6138 "22 = { , 50}, % \varepsilon
6139 "23 = { , 50}, % \vartheta
6140 "24 = { , 50}, % \varpi
6141 "25 = {100, }, % \varrho
6142 "26 = {100,100}, % \varsigma
6143 "27 = { 50, 50}, % \varphi
6144 "28 = {100,100}, % \leftharpoonup
6145 "29 = {100,100}, % \leftharpoondown
6146 "2A = {100,100}, % \rightharpoonup
6147 "2B = {100,100}, % \rightharpoondown
6148 "2C = {300,200}, % \lhook
6149 "2D = {200,300}, % \rhook
6150 "2E = { ,100}, % \triangleright
6151 "2F = {100, }, % \triangleleft
6152 "3A = { ,500}, % ., \ldotp
6153 "3B = { ,500}, % ,
6154 "3C = {200,100}, % <
6155 "3D = {300,400}, % /
6156 "3E = {100,200}, % >
6157 "3F = {200,200}, % \star
6158 "5B = { ,100}, % \flat
6159 "5E = {200,200}, % \smile
6160 "5F = {200,200}, % \frown
6161 "7C = {100, }, % \jmath
6162 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

6163 }
6164

```

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}

```

```

6165 \SetProtrusion
6166 [ name = cmr-math-symbols ]
6167 { encoding = OMS,
6168   family = cmsy,
6169   series = {m,b},
6170   shape = n }
6171 {
6172   A = {150, 50}, % \mathcal

```

```

6173     C = {    ,100},
6174     D = {    , 50},
6175     F = { 50,150},
6176     I = {    ,100},
6177     J = {100,150},
6178     K = {    ,100},
6179     L = {100,   },
6180     M = { 50, 50},
6181     N = { 50,100},
6182     P = {    , 50},
6183     Q = { 50,   },
6184     R = {    , 50},
6185     T = { 50,150},
6186     V = { 50, 50},
6187     W = {    , 50},
6188     X = {100,100},
6189     Y = {100,   },
6190     Z = {100,150},
6191     "00 = {300,300}, % -
6192     "01 = {    ,700}, % \cdot, \cdotp
6193     "02 = {150,250}, % \times
6194     "03 = {150,250}, % *, \ast
6195     "04 = {200,300}, % \div
6196     "05 = {150,250}, % \diamond
6197     "06 = {200,200}, % \pm
6198     "07 = {200,200}, % \mp
6199     "08 = {100,100}, % \oplus
6200     "09 = {100,100}, % \ominus
6201     "0A = {100,100}, % \otimes
6202     "0B = {100,100}, % \oslash
6203     "0C = {100,100}, % \odot
6204     "0D = {100,100}, % \bigcirc
6205     "0E = {100,100}, % \circ
6206     "0F = {100,100}, % \bullet
6207     "10 = {100,100}, % \asymp
6208     "11 = {100,100}, % \equiv
6209     "12 = {200,100}, % \subseteq
6210     "13 = {100,200}, % \supseteq
6211     "14 = {200,100}, % \leq
6212     "15 = {100,200}, % \geq
6213     "16 = {200,100}, % \preceq
6214     "17 = {100,200}, % \succeq
6215     "18 = {200,200}, % \sim
6216     "19 = {150,150}, % \approx
6217     "1A = {200,100}, % \subset
6218     "1B = {100,200}, % \supset
6219     "1C = {200,100}, % \ll
6220     "1D = {100,200}, % \gg
6221     "1E = {300,100}, % \prec
6222     "1F = {100,300}, % \succ
6223     "20 = {100,200}, % \leftarrow
6224     "21 = {200,100}, % \rightarrow
6225     "22 = {100,100}, % \uparrow
6226     "23 = {100,100}, % \downarrow
6227     "24 = {100,100}, % \leftrightarrows
6228     "25 = {100,100}, % \nearrow
6229     "26 = {100,100}, % \searrow
6230     "27 = {100,100}, % \simeq
6231     "28 = {100,100}, % \Leftarrow
6232     "29 = {100,100}, % \Rightarrow
6233     "2A = {100,100}, % \Uparrow
6234     "2B = {100,100}, % \Downarrow
6235     "2C = {100,100}, % \Leftrightarrow

```



```

6236 "2D = {100,100}, % \narrow
6237 "2E = {100,100}, % \swarrow
6238 "2F = { ,100}, % \propto
6239 "30 = { ,400}, % \prime
6240 "31 = {100,100}, % \infty
6241 "32 = {150,100}, % \in
6242 "33 = {100,150}, % \ni
6243 "34 = {100,100}, % \triangle, \bigtriangleup
6244 "35 = {100,100}, % \bigtriangledown
6245 "38 = { ,100}, % \forall
6246 "39 = {100, }, % \exists
6247 "3A = {200, }, % \neg
6248 "3E = {200,200}, % \top
6249 "3F = {200,200}, % \bot, \perp
6250 "5E = {100,200}, % \wedge
6251 "5F = {100,200}, % \vee
6252 "60 = { ,300}, % \vdash
6253 "61 = {300, }, % \dashv
6254 "62 = {100,100}, % \lfloor
6255 "63 = {100,100}, % \rfloor
6256 "64 = {100,100}, % \lceil
6257 "65 = {100,100}, % \rceil
6258 "66 = {150, }, % \lbrace
6259 "67 = { ,150}, % \rbrace
6260 "68 = {400, }, % \langle
6261 "69 = { ,400}, % \rangle
6262 "6C = {100,100}, % \updownarrow
6263 "6D = {100,100}, % \Updownarrow
6264 "6E = {100,300}, % \, \backslash, \setminus
6265 "72 = {100,100}, % \nabla
6266 "79 = {200,200}, % \dagger
6267 "7A = {100,100}, % \ddagger
6268 "7B = {100, }, % \mathparagraph
6269 "7C = {100,100}, % \clubsuit
6270 "7D = {100,100}, % \diamondsuit
6271 "7E = {100,100}, % \heartsuit
6272 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

6273 }
6274

```

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}
```

```

6275 </cmr>
6276 </cfg-t>

```

15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
6277 <*cfg-u>
```

Symbol font 'a'.

```

6278 <*msa>
6279 \SetProtrusion
6280 [ name = AMS-a ]
6281 { encoding = U,
6282   family = msa }

```

```

6283 {
6284   "05 = {150,250}, % \centerdot
6285   "06 = {100,100}, % \lozenge
6286   "07 = { 50, 50}, % \blacklozenge
6287   "08 = { 50, 50}, % \circlearrowright
6288   "09 = { 50, 50}, % \circlearrowleft
6289   "0A = {100,100}, % \rightleftharpoons
6290   "0B = {100,100}, % \leftrightharpoons
6291   "0D = {-50,200}, % \Vdash
6292   "0E = {-50,200}, % \Vvdash
6293   "0F = {-70,150}, % \vdash
6294   "10 = {100,150}, % \twoheadrightarrow
6295   "11 = {100,150}, % \twoheadleftarrow
6296   "12 = { 50,100}, % \leftleftarrows
6297   "13 = { 50, 80}, % \rightrightarrows
6298   "14 = {120,120}, % \uparrows
6299   "15 = {120,120}, % \downdownarrows
6300   "16 = {200,200}, % \upharpoonright
6301   "17 = {200,200}, % \downharpoonright
6302   "18 = {200,200}, % \upharpoonleft
6303   "19 = {200,200}, % \downharpoonleft
6304   "1A = { 80,100}, % \rightarrowtail
6305   "1B = { 80,100}, % \leftarrowtail
6306   "1C = { 50, 50}, % \leftrightarrows
6307   "1D = { 50, 50}, % \rightleftarrows
6308   "1E = {250,   }, % \Lsh
6309   "1F = {   ,250}, % \Rsh
6310   "20 = {100,100}, % \rightsquigarrow
6311   "21 = {100,100}, % \leftrightsquigarrow
6312   "22 = {100, 50}, % \looparrowleft
6313   "23 = { 50,100}, % \looparrowright
6314   "24 = { 50, 80}, % \circeq
6315   "25 = {   ,100}, % \succsim
6316   "26 = {   ,100}, % \gtrsim
6317   "27 = {   ,100}, % \gtrapprox
6318   "28 = {150, 50}, % \multimap
6319   "2B = {100,150}, % \doteqdot
6320   "2C = {100,150}, % \triangleq
6321   "2D = {100, 50}, % \precsim
6322   "2E = {100, 50}, % \lesssim
6323   "2F = { 50, 50}, % \lessapprox
6324   "30 = {100, 50}, % \eqslantless
6325   "31 = { 50, 50}, % \eqslantgtr
6326   "32 = {100, 50}, % \curlyeqprec
6327   "33 = { 50,100}, % \curlyeqsucc
6328   "34 = {100, 50}, % \preccurlyeq
6329   "36 = { 50,   }, % \leqslant
6330   "38 = {   , 50}, % \backprime
6331   "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
6332   "3C = { 50,100}, % \succcurlyeq
6333   "3E = {   , 50}, % \geqslant
6334   "40 = {   , 50}, % \sqsubset
6335   "41 = { 50,   }, % \sqsupset
6336   "42 = {   ,150}, % \vartriangleright, \rhd
6337   "43 = {150,   }, % \vartriangleleft, \lhd
6338   "44 = {   ,100}, % \trianglerighteq, \unrhd
6339   "45 = {100,   }, % \trianglelefteq, \unlhd
6340   "46 = {100,100}, % \bigstar
6341   "48 = { 50, 50}, % \blacktriangledown
6342   "49 = {   ,100}, % \blacktriangleright
6343   "4A = {100,   }, % \blacktriangleleft
6344   "4B = {   ,150}, % \dashrightarrow (the arrow)
6345   "4C = {150,   }, % \dashleftarrow

```

```

6346 "4D = { 50, 50}, % \vartriangle
6347 "4E = { 50, 50}, % \blacktriangle
6348 "4F = { 50, 50}, % \triangledown
6349 "50 = { 50, 50}, % \eqcirc
6350 "56 = { ,150}, % \Rrightarrow
6351 "57 = {150, }, % \Lleftarrow
6352 "58 = {100,300}, % \checkmark
6353 "5C = { 50, 50}, % \angle
6354 "5D = { 50, 50}, % \measuredangle
6355 "5E = { 50, 50}, % \sphericalangle
6356 "5F = { , 50}, % \varpropto
6357 "60 = {100,100}, % \smallsmile
6358 "61 = {100,100}, % \smallfrown
6359 "62 = { 50, }, % \Subset
6360 "63 = { , 50}, % \Supset
6361 "66 = {150,150}, % \curlywedge
6362 "67 = {150,150}, % \curlyvee
6363 "68 = { 50,150}, % \leftthreetimes
6364 "69 = {100, 50}, % \rightthreetimes
6365 "6C = { 50, 50}, % \bumpeq
6366 "6D = { 50, 50}, % \Bumpeq
6367 "6E = {100, }, % \lll
6368 "6F = { ,100}, % \ggg
6369 "70 = { 50,100}, % \ulcorner
6370 "71 = {100, 50}, % \urcorner
6371 "75 = {150,200}, % \dotplus
6372 "76 = { 50,100}, % \backsim
6373 "78 = { 50,100}, % \llcorner
6374 "79 = {100, 50}, % \lrcorner
6375 "7C = {100,100}, % \intercal
6376 "7D = { 50, 50}, % \circledcirc
6377 "7E = { 50, 50}, % \circledast
6378 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

6379 }
6380
6381 </msa>

```

Symbol font 'b'.

```

6382 <*msb>
6383 \SetProtrusion
6384 [ name = AMS-b ]
6385 { encoding = U,
6386   family = msb }
6387 {
6388   A = { 50, 50}, % \mathbb
6389   C = { 50, 50},
6390   G = { , 50},
6391   L = { , 50},
6392   P = { , 50},
6393   R = { , 50},
6394   T = { , 50},
6395   V = { 50, 50},
6396   X = { 50, 50},
6397   Y = { 50, 50},
6398   "00 = { 50, 50}, % \lvertneqq
6399   "01 = { 50, 50}, % \gvertneqq
6400   "02 = { 50, 50}, % \nleq
6401   "03 = { 50, 50}, % \ngeq
6402   "04 = {100, 50}, % \nless
6403   "05 = { 50,150}, % \ngtr
6404   "06 = {100, 50}, % \nprec

```

```

6405 "07 = { 50,150}, % \nsucc
6406 "08 = { 50, 50}, % \lneqq
6407 "09 = { 50, 50}, % \gneqq
6408 "0A = {100,100}, % \lneqslant
6409 "0B = {100,100}, % \ngeqslant
6410 "0C = {100, 50}, % \lneq
6411 "0D = { 50,100}, % \gneq
6412 "0E = {100, 50}, % \npreceq
6413 "0F = { 50,100}, % \nsucceq
6414 "10 = { 50,  }, % \precnsim
6415 "11 = { 50, 50}, % \succnsim
6416 "12 = { 50, 50}, % \lnsim
6417 "13 = { 50, 50}, % \gnsim
6418 "14 = { 50, 50}, % \lneqq
6419 "15 = { 50, 50}, % \ngeqq
6420 "16 = { 50, 50}, % \precneqq
6421 "17 = { 50, 50}, % \succneqq
6422 "18 = { 50, 50}, % \precnapprox
6423 "19 = { 50, 50}, % \succnapprox
6424 "1A = { 50, 50}, % \lnapprox
6425 "1B = { 50, 50}, % \gnapprox
6426 "1C = {150,200}, % \nsim
6427 "1D = { 50, 50}, % \ncong
6428 "1E = {100,150}, % \diagup
6429 "1F = {100,150}, % \diagdown
6430 "20 = {100, 50}, % \varsubsetneq
6431 "21 = { 50,100}, % \varsupsetneq
6432 "22 = {100, 50}, % \subsetneqq
6433 "23 = { 50,100}, % \supsetneqq
6434 "24 = {100, 50}, % \subsetneqq
6435 "25 = { 50,100}, % \supsetneqq
6436 "26 = {100, 50}, % \varsubsetneqq
6437 "27 = { 50,100}, % \varsupsetneqq
6438 "28 = {100, 50}, % \subsetneq
6439 "29 = { 50,100}, % \supsetneq
6440 "2A = {100, 50}, % \subseteq
6441 "2B = { 50,100}, % \supseteq
6442 "2C = { 50,100}, % \nparallel
6443 "2D = {100,150}, % \nmid
6444 "2E = {150,150}, % \nshortmid
6445 "2F = {100,100}, % \nshortparallel
6446 "30 = {  ,150}, % \nvDash
6447 "31 = {  ,150}, % \nVDash
6448 "32 = {  ,100}, % \nvDash
6449 "33 = {  ,100}, % \nVDash
6450 "34 = {  ,100}, % \ntrianglerighteq
6451 "35 = {100,  }, % \triangleleft
6452 "36 = {100,  }, % \triangleleft
6453 "37 = {  ,100}, % \triangleright
6454 "38 = {100,200}, % \leftarrow
6455 "39 = {100,200}, % \rightarrow
6456 "3A = {100,100}, % \Leftarrow
6457 "3B = { 50,100}, % \Rightarrow
6458 "3C = {100,100}, % \Leftrightarrow
6459 "3D = {100,200}, % \leftrightharpoonup
6460 "3E = { 50, 50}, % \divertimes
6461 "3F = { 50, 50}, % \varnothing
6462 "60 = {200,  }, % \Finv
6463 "61 = {  , 50}, % \Game
6464 "68 = {100,100}, % \eqsim
6465 "69 = { 50,  }, % \beth
6466 "6A = { 50,  }, % \gimel
6467 "6B = {150,  }, % \daleth

```

```

6468 "6C = {200, }, % \lessdot
6469 "6D = { ,200}, % \gtrdot
6470 "6E = {100,200}, % \ltimes
6471 "6F = {150,100}, % \rtimes
6472 "70 = { 50,100}, % \shortmid
6473 "71 = { 50, 50}, % \shortparallel
6474 "72 = {200,300}, % \smallsetminus
6475 "73 = {100,200}, % \thicksim
6476 "74 = { 50,100}, % \thickapprox
6477 "75 = { 50, 50}, % \approxeq
6478 "76 = { 50,100}, % \succapprox
6479 "77 = { 50, 50}, % \precapprox
6480 "78 = {100,100}, % \curvearrowleft
6481 "79 = { 50,150}, % \curvearrowright
6482 "7A = { 50,200}, % \digamma
6483 "7B = {100, 50}, % \varkappa
6484 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

6485 }
6486
6487 (/msb)

```

15.8.8 Euler

Euler Roman font (package `euler`).

```

6488 (*eur)
6489 \SetProtrusion
6490 [ name = euler ]
6491 { encoding = U,
6492   family = eur }
6493 {
6494   "01 = {100,100},
6495   "03 = {100,150},
6496   "06 = { ,100},
6497   "07 = {100,150},
6498   "08 = {100,100},
6499   "0A = {100,100},
6500   "0B = { , 50},
6501   "0C = { ,100},
6502   "0D = {100,100},
6503   "0E = { ,100},
6504   "0F = {100,100},
6505   "10 = {100,100},
6506   "13 = { ,100},
6507   "14 = { ,100},
6508   "15 = { , 50},
6509   "16 = { , 50},
6510   "17 = { 50,100},
6511   "18 = { 50,100},
6512   "1A = { , 50},
6513   "1B = { , 50},
6514   "1C = { 50,100},
6515   "1D = { 50,100},
6516   "1E = { 50,100},
6517   "1F = { 50,100},
6518   "20 = { , 50},
6519   "21 = { , 50},
6520   "22 = { 50,100},
6521   "24 = { , 50},
6522   "27 = { 50,100},

```

```

6523     1 = {100,100},
6524     7 = { 50,100},
6525    "3A = {300,500},
6526    "3B = {200,400},
6527    "3C = {200,100},
6528    "3D = {200,200},
6529    "3E = {100,200},
6530     A = {  ,100},
6531     D = {  , 50},
6532     J = { 50,  },
6533     K = {  , 50},
6534     L = {  , 50},
6535     Q = {  , 50},
6536     T = { 50,  },
6537     X = { 50, 50},
6538     Y = { 50,  },
6539     h = {  , 50},
6540     k = {  , 50}
6541   }
6542

```

Extended by the `eulervm` package.

```

6543 \SetProtrusion
6544 [ name = euler-vm,
6545   load = euler ]
6546 { encoding = U,
6547   family = zeur }
6548 {
6549   "28 = {100,200},
6550   "29 = {100,200},
6551   "2A = {100,150},
6552   "2B = {100,150},
6553   "2C = {200,300},
6554   "2D = {200,300},
6555   "2E = {  ,100},
6556   "2F = {100,  },
6557   "3F = {150,150},
6558   "5B = {  ,100},
6559   "5E = {100,100},
6560   "5F = {100,100},
6561   "80 = {  , 50},
6562   "81 = {200,250},
6563   "82 = {100,200}
6564 }
6565
6566 </eur>

```

Euler Script font (`euca1`).

```

6567 <*eus>
6568 \SetProtrusion
6569 [ name = euscript ]
6570 { encoding = U,
6571   family = eus }
6572 {
6573   A = {100,100},
6574   B = { 50,100},
6575   C = { 50, 50},
6576   D = { 50,100},
6577   E = { 50,100},
6578   F = { 50,  },
6579   G = { 50,  },
6580   H = {  ,100},
6581   K = {  , 50},

```

```

6582     L = {    ,150},
6583     M = {    , 50},
6584     N = {    , 50},
6585     O = { 50, 50},
6586     P = { 50, 50},
6587     T = {    ,100},
6588     U = {    , 50},
6589     V = { 50, 50},
6590     W = { 50, 50},
6591     X = { 50, 50},
6592     Y = { 50,   },
6593     Z = { 50,100},
6594     "00 = {250,250},
6595     "18 = {200,200},
6596     "3A = {200,150},
6597     "40 = {    ,100},
6598     "5E = {100,100},
6599     "5F = {100,100},
6600     "66 = { 50,   },
6601     "67 = {    , 50},
6602     "6E = {200,200}
6603 }
6604
6605 \SetProtrusion
6606 [ name    = euscript-vm,
6607   load    = euscript ]
6608 { encoding = U,
6609   family   = zeus   }
6610 {
6611   "01 = {600,600},
6612   "02 = {200,200},
6613   "03 = {200,200},
6614   "04 = {200,200},
6615   "05 = {150,150},
6616   "06 = {200,200},
6617   "07 = {200,200},
6618   "08 = {100,100},
6619   "09 = {100,100},
6620   "0A = {100,100},
6621   "0B = {100,100},
6622   "0C = {100,100},
6623   "0D = {100,100},
6624   "0E = {150,150},
6625   "0F = {100,100},
6626   "10 = {150,150},
6627   "11 = {100,100},
6628   "12 = {150,100},
6629   "13 = {100,150},
6630   "14 = {150,100},
6631   "15 = {100,150},
6632   "16 = {200,100},
6633   "17 = {100,200},
6634   "19 = {150,150},
6635   "1A = {150,100},
6636   "1B = {100,150},
6637   "1C = {100,100},
6638   "1D = {100,100},
6639   "1E = {250,100},
6640   "1F = {100,250},
6641   "20 = {150,200},
6642   "21 = {150,200},
6643   "22 = {150,150},
6644   "23 = {150,150},

```

```

6645 "24 = {100,200},
6646 "25 = {150,150},
6647 "26 = {150,150},
6648 "27 = {100,100},
6649 "28 = {100,100},
6650 "29 = {100,150},
6651 "2A = {100,100},
6652 "2B = {100,100},
6653 "2C = {100,100},
6654 "2D = {150,150},
6655 "2E = {150,150},
6656 "2F = {100,100},
6657 "30 = {100,100},
6658 "31 = {100,100},
6659 "32 = {100,100},
6660 "33 = {100,100},
6661 "34 = {100,100},
6662 "35 = {100,100},
6663 "3E = {150,150},
6664 "3F = {150,150},
6665 "60 = { ,200},
6666 "61 = {200, },
6667 "62 = {100,100},
6668 "63 = {100,100},
6669 "64 = {100,100},
6670 "65 = {100,100},
6671 "68 = {300, },
6672 "69 = { ,300},
6673 "6C = {100,100},
6674 "6D = {100,100},
6675 "6F = {100,100},
6676 "72 = {100,100},
6677 "73 = {200,100},
6678 "76 = { ,100},
6679 "77 = {100, },
6680 "78 = { 50, 50},
6681 "79 = {100,100},
6682 "7A = {100,100},
6683 "7D = {150,150},
6684 "7E = {100,100},
6685 "A8 = {100,100},
6686 "A9 = {100,100},
6687 "AB = {200,200},
6688 "BA = { ,200},
6689 "BB = { ,200},
6690 "BD = {200,200},
6691 "DE = {200,200}
6692 }
6693
6694 </eus>

```

Euler Fraktur font (eufrak).

```

6695 <*euf>
6696 \SetProtrusion
6697 [ name = mathfrak ]
6698 { encoding = U,
6699   family = euf }
6700 {
6701   A = { , 50},
6702   B = { , 50},
6703   C = { 50, 50},
6704   D = { , 80},
6705   E = { 50, },

```



```

6706      G = {      , 50},
6707      L = {      , 80},
6708      O = {      , 50},
6709      T = {      , 80},
6710      X = { 80, 50},
6711      Z = { 80, 50},
6712      b = {      , 50},
6713      c = {      , 50},
6714      k = {      , 50},
6715      p = {      , 50},
6716      q = { 50,    },
6717      v = {      , 50},
6718      w = {      , 50},
6719      x = {      , 50},
6720      1 = {100,100},
6721      2 = { 80, 80},
6722      3 = { 80, 50},
6723      4 = { 80, 50},
6724      7 = { 50, 50},
6725      "12 = {500,500},
6726      "13 = {500,500},
6727      ! = {      ,200},
6728      ' = {200,300},
6729      ( = {200,    },
6730      ) = {      ,200},
6731      * = {200,200},
6732      + = {200,250},
6733      - = {200,200},
6734      {,} = {300,300},
6735      . = {400,400},
6736      {=} = {200,200},
6737      : = {      ,200},
6738      ; = {      ,200},
6739      ] = {      ,200}
6740  }
6741
6742  </euf>
6743  </cfg-u>

```

15.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym²¹).

```

6744  <*cfg-e>
6745  \SetProtrusion
6746  <zpeu|euroitc> { encoding = U,
6747  <mvs> { encoding = {OT1,U},
6748  <zpeu> family = zpeu }
6749  <euroitc> family = {euroitc,euroitcs} }
6750  <mvs> family = mvs }
6751  {
6752  <zpeu> E = {50, }
6753  <euroitc> E = {100,50}
6754  <mvs> 164 = {50,50}, % \EUR
6755  <mvs> 068 = {50,-100} % \EURdig
6756  }
6757
6758  <*zpeu|euroitc>
6759  \SetProtrusion

```

21 Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

Figure 1: Example for interword spacing (from Siemoneit 1989). The numbers indicate the preference/order when the interword space needs to be shrunk.

2 6 7 5 3 4 1

Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei

```

6760 { encoding = U,
6761 (zpeu) family = zpeu,
6762 (euroitc) family = {euroitc,euroitcs},
6763 shape = it* }
6764 {
6765 (zpeu) E = {100,-50}
6766 (euroitc) E = {100,}
6767 }
6768
6769 (/zpeu|euroitc)
6770 (*zpeu)
6771 \SetProtrusion
6772 { encoding = U,
6773 family = {zpeus,eurosans} }
6774 {
6775 E = {100,50}
6776 }
6777
6778 \SetProtrusion
6779 { encoding = U,
6780 family = {zpeus,eurosans},
6781 shape = it* }
6782 {
6783 E = {200, }
6784 }
6785
6786 (/zpeu)
6787 (/cfg-e)

```

15.9 Interword spacing

Default unit is space.

```

6788 (*m-t)
6789 %%% -----
6790 %%% INTERWORD SPACING
6791
6792 \SetExtraSpacing
6793 [ name = default ]
6794 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6795 {

```

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

6796 {,} = { , -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)

6797 r = { , -300, 300},

- [before or] after lowercase characters with ascenders

6798 b = { , -200, 200},

6799 d = { , -200, 200},

6800 f = { , -200, 200},

6801 h = { , -200, 200},

6802 k = { , -200, 200},

6803 l = { , -200, 200},

6804 t = { , -200, 200},

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

6805 c = { , -100, 100},

6806 p = { , -100, 100},

6807 v = { , -100, 100},

6808 w = { , -100, 100},

6809 z = { , -100, 100},

6810 x = { , -100, 100},

6811 y = { , -100, 100},

- [before or] after lowercase characters with x-height plus descender without additional optical space

6812 i = { , 50, -50},

6813 m = { , 50, -50},

6814 n = { , 50, -50},

6815 u = { , 50, -50},

- after colon and semicolon

6816 : = { , 200, -200},

6817 ; = { , 200, -200},

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

6818 . = { , 250, -250},

6819 ! = { , 250, -250},

6820 ? = { , 250, -250}

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

6821 }

6822

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
\stbrcode\font`t=-50
test test
\bye
```

Some more characters in T2A.²²

```
6823 <*/m-t>
6824 \SetExtraSpacing
6825   [ name      = T2A,
6826     load      = default ]
6827   { encoding = T2A,
6828     family   = cmr }
6829   {
6830     \cyrgh = { , -300, 300 },
6831     \cyrb = { , -200, 200 },
6832     \cyrk = { , -200, 200 },
6833     \cyrh = { , -100, 100 },
6834     \cyrp = { , -100, 100 },
6835     \cyrq = { , -100, 100 },
6836     \cyrj = { , -100, 100 },
6837     \cyrz = { , 50, -50 },
6838     \cyrp = { , 50, -50 },
6839     \cyrj = { , 50, -50 },
6840     \cyrshrt = { , 50, -50 },
6841   }
6842
6843 </m-t>
```

15.9.1 Nonfrenchspacing

The following settings simulate `\nonfrenchspacing` (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the *TeXbook*:

‘If the space factor f is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [...] Then the stretch component is multiplied by $f/1000$, while the shrink component is multiplied by $1000/f$.’

The ‘extra space’ (`\fontdimen7`) for Computer Modern Roman is a third of `\fontdimen2`, i.e., 333.

```
6844 \SetExtraSpacing
6845   [ name      = nonfrench-cmr,
6846     load      = default,
6847     context   = nonfrench ]
6848   { encoding = {OT1,T1,LY1,OT4,QX,T5},
```

22 Contributed by *Karl Karlsson*.

```
6849     family    = cmr }
6850     {
```

latex.ltx has:

```
\def\nonfrenchspacing{
  \sfcode`. 3000
  \sfcode`\? 3000
  \sfcode`\! 3000
```

```
6851     . = {333,2000,-667},
6852     ? = {333,2000,-667},
6853     ! = {333,2000,-667},
```

```
\sfcode`\: 2000
```

```
6854     : = {333,1000,-500},
```

```
\sfcode`\; 1500
```

```
6855     ; = {    , 500,-333},
```

```
\sfcode`\, 1250
```

```
6856     {,}= {    , 250,-200}
```

```
}
```

```
6857   }
6858
```

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.

```
6859 \SetExtraSpacing
6860   [ name      = nonfrench-default,
6861     load      = default,
6862     context   = nonfrench ]
6863   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6864   {
6865     . = {240,2000,-667},
6866     ? = {240,2000,-667},
6867     ! = {240,2000,-667},
6868     : = {240,1000,-500},
6869     ; = {    , 500,-333},
6870     {,}= {    , 250,-200}
6871   }
6872
```

15.10 Additional kerning

Default unit is 1 em.

```
6873 %%% -----
6874 %%% ADDITIONAL KERNING
6875
```

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

```
6876 \SetExtraKerning
6877   [ name = empty ]
```

```

6878 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
6879 { }
6880

```

15.10.1 French

The ratio of `\fontdimen 2` to `\fontdimen 6` varies for different fonts, so that either the kerning of the colon (which should be a space, i. e., `\fontdimen 2`) or that of the other punctuation characters (T_EX's `\thinspace`, i. e., one sixth of `\fontdimen 6`) may be inaccurate, depending on which unit we choose (space or 1em). For Times, for example, a thin space would be 665. I don't know whether French typography really wants a thin space, or rather (as it happens to turn out with CMR) half a space. (Wikipedia²³ 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.)

```

6881 \SetExtraKerning
6882 [ name      = french-default,
6883   context   = french,
6884   unit      = space ]
6885 { encoding = {OT1,T1,LY1} }
6886 {
6887   : = {1000,}, % = \fontdimen2
6888   ; = {500, }, % ~ \thinspace
6889   ! = {500, },
6890   ? = {500, }
6891 }
6892

```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfT_EX.

```

6893 \SetExtraKerning
6894 [ name      = french-guillemets,
6895   context   = french-guillemets,
6896   load      = french-default,
6897   unit      = space ]
6898 { encoding = {T1,LY1} }
6899 {
6900   \guillemotleft = { ,800}, % = 0.8\fontdimen2
6901   \guillemotright = {800, }
6902 }
6903
6904 \SetExtraKerning
6905 [ name      = french-guillemets-OT1,
6906   context   = french-guillemets,
6907   load      = french-default,
6908   unit      = space ]
6909 { encoding = OT1 }
6910 { }
6911

```

15.10.2 Turkish

```

6912 \SetExtraKerning
6913 [ name      = turkish,
6914   context   = turkish ]
6915 { encoding = {OT1,T1,LY1} }
6916 {

```

23 http://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.

```

6917      : = {167, }, % = \thinspace
6918      ! = {167, },
6919      {=} = {167, }
6920      }
6921
6922 </m-t>
6923 </config>

```

16 Auxiliary file for micro fine tuning

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

```

6924 <*test>
6925 \documentclass{article}
6926
6927 %% Here you can specify the font you want to test, using
6928 %% the commands \fontfamily, \fontseries and \fontshape.
6929 %% Make sure to end all lines with a comment character!
6930 \newcommand*\TestFont{%
6931   \fontfamily{ppl}%
6932   %% \fontseries{b}%
6933   %% \fontshape{it}% sc, sl
6934 }
6935
6936 \usepackage{ifthen}
6937 \usepackage[T1]{fontenc}
6938 \usepackage[latin1]{inputenc}
6939 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
6940
6941 \pagestyle{empty}
6942 \setlength{\parindent}{0pt}
6943 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
6944 \newcommand*\testprotrusion[2][\]{%
6945   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
6946   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6947   you know the rest%
6948   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6949   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6950   \ifthenelse{\equal{#1}{l}}{\crulefill}{\leftarrowfill}
6951   \linebreak
6952   {\fontencoding{\encodingdefault}%
6953    \fontseries{\seriesdefault}%
6954    \fontshape{\shapedefault}%
6955    \selectfont
6956    Here is the beginning of a line, \dotfill and here is its end}\linebreak
6957 }
6958 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
6959 \def\stripprefix#1>{}
6960 \newcount\charcount
6961 \begin{document}
6962
6963 \microtypesetup{expansion=false}
6964
6965 {\centering The font in this document is called by:\\
6966 \texttt{\showTestFont}\par}\bigskip
6967
6968 \TestFont\selectfont
6969 This line intentionally left empty\linebreak
6970 %% A -- Z
6971 \charcount=65
6972 \loop
6973   \testprotrusion{\char\charcount}
6974   \advance\charcount 1

```

```

6975 \ifnum\charcount < 91 \repeat
6976 %% a -- z
6977 \charcount=97
6978 \loop
6979 \testprotrusion{\char\charcount}
6980 \advance\charcount 1
6981 \ifnum\charcount < 123 \repeat
6982 %% 0 -- 9
6983 \charcount=48
6984 \loop
6985 \testprotrusion{\char\charcount}
6986 \advance\charcount 1
6987 \ifnum\charcount < 58 \repeat
6988 %%
6989 \testprotrusion[r]{,}
6990 \testprotrusion[r]{.}
6991 \testprotrusion[r]{;}
6992 \testprotrusion[r]{:}
6993 \testprotrusion[r]{?}
6994 \testprotrusion[r]{!}
6995 \testprotrusion[l]{\textexclamdown}
6996 \testprotrusion[l]{\textquestiondown}
6997 \testprotrusion[r]{\}}
6998 \testprotrusion[l]{\{ }
6999 \testprotrusion{/}
7000 \testprotrusion{\char`\}
7001 \testprotrusion{-}
7002 \testprotrusion{\textendash}
7003 \testprotrusion{\textemdash}
7004 \testprotrusion{\textquoteleft}
7005 \testprotrusion{\textquoteright}
7006 \testprotrusion{\textquotedblleft}
7007 \testprotrusion{\textquotedblright}
7008 \testprotrusion{\quotesinglbase}
7009 \testprotrusion{\quotedblbase}
7010 \testprotrusion{\guilsinglleft}
7011 \testprotrusion{\guilsinglright}
7012 \testprotrusion{\guillemotleft}
7013 \testprotrusion{\guillemotright}
7014
7015 \newpage
7016 The following displays the current font stretched by 5%,
7017 normal, and shrunk by 5\%:
7018
7019 \bigskip
7020 \newlength{\MTln}
7021 \newcommand*{\teststring}
7022 {ABCDEFGHIJKLMNQRSTUvwxyz0123456789}
7023 \settowidth{\MTln}{\teststring}
7024 \microtypesetup{expansion=true}
7025
7026 \parbox{1.05\MTln}{\teststring\linebreak\}
7027 \teststring\par\bigskip
7028 \parbox{0.95\MTln}{\teststring}
7029
7030 \end{document}
7031 /test

```

Needless to say that things may always be improved. For suggestions, mail to w.m.l@gmx.net.

A Change history

Version 1.0 (2004/09/11)

General: Initial version 1

Version 1.1 (2004/09/21)

General: configuration file names in lowercase (suggested by <i>Harald Harders</i>)	80	<code>\MT@get@basefamily</code> : only remove suffix if it is ‘x’ or ‘j’	81
issue an error instead of a warning, when pdfTeX version is too old for autoexpand	123	<code>\MT@get@listname@</code> : don’t check for empty attributes list	81
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i>)	131	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i>)	43
Protrusion: add factors for some more characters settings for Adobe Minion (contributed by <i>Harald Harders</i>)	137	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i>)	85
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance	139	<code>\MT@pdftex@no</code> : fix: version check (reported by <i>Harald Harders</i>)	38
<code>\MT@declare@sets</code> : remove spaces around set name	107	<code>\MT@permute</code> : don’t use sets for empty encoding ..	109
<code>\MT@DeclareSet</code> : remove spaces around first argument	94	<code>\MT@split@codes</code> : fix: allow zero and negative values ..	59
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded ..	94	<code>\MT@use@set</code> : remove spaces around set name	98
	80	<code>\UseMicrotypeSet</code> : remove spaces around first argument	98

Version 1.2 (2004/10/03)

Font sets: declare cmr as an alias of cmr	129	changed	95
new: allmath and basicmath	128	<code>\MT@get@inh@list</code> : fix: set inheritance list \globally to \empty	83
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding	161	<code>\MT@get@listname@</code> : alternatively check for alias font name	81
add settings for Computer Modern Roman math symbols	166	<code>\MT@get@size</code> : additional magic to catch some errors hijack \set@fontsize instead of \set@fontsize ..	97
<code>\MT@encoding@check</code> : check whether only one encoding specified	108	<code>\MT@loop</code> : fix: new macro, used instead of \loop ..	47
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement	55	<code>\MT@maybe@do</code> : also check for alias font name	55
<code>\MT@get@basefamily</code> : also remove ‘w’ (swash capitals)	81	<code>\MT@permute@@@@@</code> : more sanity checks for \SetProtrusion and \SetExpansion	110
<code>\MT@get@highlevel</code> : check whether defaults have		<code>\MT@setupfont</code> : also search for alias font file	53
		fix: call \@@enc@update if necessary	53

Version 1.3 (2004/10/27)

General: fix: specifying load option does no longer require to give a name, too	104	<code>\MT@fix@catcode</code> : check some category codes (compatibility with german)	34
Font sets: declare aer, zer and hfor as aliases of cmr	129	<code>\MT@load@list</code> : check whether list exists	79

Version 1.4 (2004/11/12)

General: check for pdfcprot	51	(OT1, T1, lmr)	143
don’t use scratch registers in global definitions ..	84	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options	118
use \pickup@font instead of \define@newfont as the hook for \MT@setupfont	89	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too	101
use one instead of five counters	47		
Protrusion: tweak quote characters for cmr variants			

Version 1.4a (2004/11/17)

General: new option: final	115	when reading files (reported by <i>Michael Hoppe</i>)	80
\MT@cfg@catcodes: fix: reset some more catcodes			

Version 1.4b (2004/11/26)

General: fix: set catcodes before reading global configuration file (reported by <i>Christoph Bier</i>) . .	117	name if encoding failed	56
new message if \pdfoutput is changed	121	\MT@get@basefamily: fix: failed for font names of the form abczz (reported by <i>Georg Verwey</i>)	81
optimisation: use less \expandafters and \csnames	41	\MT@get@slot: don't define \MT@char globally (save stack problem)	84
Protrusion: harmonise dashes in upshape and italic (cmr, pad, ppl)	137	\MT@ifdimen: don't set \MT@count globally (save stack problem)	44
slanted like italics	147	\MT@use@set: don't use undeclared font sets	98
\MT@checklist@family: fix: don't try alias family			

Version 1.5 (2004/12/15)

General: defaults: step: 4 (suggested by <i>Hàn Thê Thành</i>)	116	\MT@cfg@catcodes: reset catcode of '=' (compatibility with Turkish babel)	80
defaults: calculate step as min(stretch,shrink)/5	122	\MT@fix@catcode: reset catcode of '~' (compatibility with chemsym)	34
defaults: turn off expansion for DVI output	121	\MT@get@highlevel: don't test defaults if called after begin document	95
disable automatic expansion for DVI output . . .	123	\MT@scale@factor: warning for factors outside limits	61
new option: selected, by default false (suggested by <i>Hàn Thê Thành</i>)	114	\MT@scale@to@em: don't use \lpcode and \rprcode for the calculation	59
Documentation: add 'Short history'	29	\MT@set@ex@codes: allow non-selected font expansion	65
add note about DVIoutput option	9	\MT@set@pr@codes: adjust protrusion factors before setting the inheriting characters	57
Inheritance: remove \ss from T1 list, add \DJ . . .	131		
Protrusion: settings for Bitstream Charter	138		
\DeclareMicrotypeAlias: remove spaces around arguments	100		

Version 1.6 (2005/01/24)

General: defaults: turn off expansion for old pdfTeX versions	116	improve settings for numbers (pointed out by <i>Peter Muthesius</i>)	140
disable automatic expansion for old pdfTeX versions	123	tune CMR math letters (OML encoding)	166
load a font if none is selected	52	\MT@get@charwd: use e-TeX's \fontcharwd, if available	60
new option: factor, by default 1000	116	\MT@get@inh@list: correct message if selected is false	83
restructure dtx file	128	\MT@set@ex@codes: introduce factor option	65
test whether \pickup@font has changed	91	\MT@set@pr@codes: introduce factor option	57
test whether numeric options receive a number	116	\MT@use@set: retain current set if new set is undeclared	98
use e-TeX's \ifcsname and \ifdefined if defined	42	\MT@vinfo: new macro instead of \ifMT@verbose . .	35
Protrusion: add italic uppercase Greek letters . .	147		

Version 1.6a (2005/02/02)

Documentation: add table of fonts with tailored protrusion settings	21	reported by <i>Bernard Gaulle</i>)	84
\MT@get@slot: completely redone, hopefully more robust (compatible with frenchpro; problem		\MT@pdfTeX@no: new macro	37
		\MT@reset@ef@codes: only reset \efcodes for older pdfTeX versions	65

Version 1.7 (2005/03/23)

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i>)	95	fix: remove space after autoexpand	107
disallow automatic expansion if pdfTeX too old	107	modify \showhyphens	124
		new value for verbose option: errors	115

shorter command names	48	test whether <code>\(encoding)\(...</code> is defined	84
warning when running in draft mode	121	<code>\MT@if@list@exists</code> : don't define <code>\MT@#1@c@name</code>	
Documentation: add hint about compatibility	27	globally, here and elsewhere	83
remove table of match order	12	<code>\MT@if@dimen</code> : comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i>)	44
Protrusion: fix: remove <code>\</code> from OT1, add <code>\textbackslash</code> to T1 encoding	141	<code>\MT@increment</code> : use e-TeX's <code>\numexpr</code> if available	47
<code>\DeclareMicrotypeAlias</code> : may also be used inside configuration files	100	<code>\MT@is@composite</code> : new macro: construct command for composite character; no uncontrolled expansion	87
<code>\LoadMicrotypeFile</code> : new command (suggested by <i>Andreas Böhmann</i>)	100	<code>\MT@scale</code> : new macro: use e-TeX's <code>\numexpr</code> if available	47
<code>\Microtype@Hook</code> : new command for font package authors	117	<code>\MT@set@ex@codes</code> : two versions of this macro	65
<code>\microtypesetup</code> : fix: warning also when setting to (no)compatibility	118	<code>\MT@split@name</code> : don't define <code>\MT@encoding</code> &c. globally	95
<code>\MT@begin@catcodes</code> : also use inside configuration commands	81	<code>\MT@test@ast</code> : make it simpler	95
<code>\MT@cfg@catcodes</code> : reset catcode of <code>'</code> (compatibility with french* packages)	80	<code>\MT@try@order</code> : always check for size, too (suggested by <i>Andreas Böhmann</i>)	82
<code>\MT@get@listname@</code> : use <code>\etfor</code> (<i>Andreas Böhmann's</i> idea)	81	fix: also check for <code>//(series)/(shape)//</code> (reported by <i>Andreas Böhmann</i>)	82
<code>\MT@get@slot</code> : remove backslash hack	84	<code>\MT@warn@code@too@large</code> : new macro: type out maximum protrusion factor	61
test for <code>\chardef</code> commands	84	<code>\MT@warn@err</code> : new macro: for verbose=errors	35

Version 1.8 (2005/06/23)

General: <code>\SetProtrusion</code> : new key: unit	106	<code>\MT@find@file</code> : no longer wrap names in commands	80
if font substitution has occurred, set up the substitute font, not the selected one	89	<code>\MT@get@charwd</code> : warning for missing (resp. zero-width) characters	60
new option: config to load a different main configuration file	116	<code>\MT@get@font@dimen@six</code> : new macro: test whether <code>\fontdimen6</code> is defined	58
new option: unit, by default character	116	<code>\MT@get@listname@</code> : made recursive	81
Documentation: add example for factor option	13	<code>\MT@get@slot</code> : fix: expand active characters	84
add example of how to get rid of a widow (suggested by <i>Adam Kucharczyk</i>)	15	test whether <code>\(encoding)\(...</code> is defined made more robust	84
add hint about error messages	27	<code>\MT@get@unit</code> : new macro: get unit for codes	62
Font sets: add U encoding to allmath	128	<code>\MT@in@list</code> : made recursive	46
declare <code>pxr</code> and <code>txr</code> as aliases of <code>ppl</code> resp. <code>ptm</code>	130	<code>\MT@is@active</code> : new macro: translate inputenc-defined characters	86
Inheritance: remove <code>\DJ</code> from T1 list (it's the same as <code>\DH</code>)	131	<code>\MT@is@letter</code> : warning for non-ASCII characters	85
Protrusion: add LY1 characters for Times	146	<code>\MT@led@kern</code> : character protrusion with <code>ledmac</code>	49
settings for AMS math fonts	169	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\@for</code>	45
verified settings for slanted Computer Modern Roman	154	<code>\MT@map@tlist@n</code> : new macro: used instead of <code>\@tfor</code>	46
<code>\add@accent</code> : fix: disable micro-typographic setup inside <code>\add@accent</code> (reported by <i>Stephan Hennig</i>)	91	<code>\MT@old@cmd</code> : renamed commands from <code>\..MicroType..</code> to <code>\..Microtype..</code>	35
<code>\DeclareMicrotypeAlias</code> : warning when overriding an alias font	100	<code>\MT@pdftex@no</code> : case 5: pdfTeX 1.30	37
<code>\DeclareMicrotypeSetDefault</code> : new command: set default font set	99	<code>\MT@permute@@@@@</code> : add ranges to the beginning of the lists	110
<code>\MT@cfg@catcodes</code> : reset catcodes of the remaining ASCII characters	80	<code>\MT@scale</code> : fix: remove spaces in e-TeX variant (reported by <i>Mark Rossi</i>)	47
<code>\MT@check@rlist</code> : made recursive	111	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when hyperref is loaded	50
<code>\MT@curr@list@name</code> : new macro: current list type and name	88	restore csquotes's active characters	50
<code>\MT@declare@sets</code> : warning when redefining a set	94	restore percent character if Spanish babel is loaded	50
<code>\MT@define@set@key@</code> : use comma lists instead of token lists	94	<code>\MT@split@codes</code> : get character width once only	59
		<code>\MT@use@set</code> : fix: remove braces in first line	98
		<code>\MT@xadd</code> : simplified	45

Version 1.9 (2005/10/28)

General: <code>\DeclareMicrotypeSet</code> : new key: font	97	<code>\SetProtrusion</code> : value 'relative' renamed to	
--	----	---	--

‘character’ for key unit	106	settings for T5 encoded Computer Modern Roman	137
allow context-specific font setup	89	\DisableLigatures: new command: disable ligatures (requires pdf \TeX 1.30)	100
disable expansion if both step and shrink are zero	123	\microtypecontext: new command: change setup context in the document	92
disable microtype setup inside hyperref’s \pdfstringdef (reported by <i>Hàn Thế Thành</i>)	51	\MT@checklist@family: fix: add two missing \expandafters	56
fix: use true as the default value	113	\MT@detokenize@c: fix the \mathcal{E} - \TeX version	43
option unit: rename value relative to character	116	\MT@exp@two@n: new macros: less \expandafters	42
warning if user requested zero step	122	\MT@get@opt: new key ‘preset’ to set all characters to the specified value before loading the lists	62
Documentation: add hint about verbatim environment	26	\MT@is@active: redone: use \set@display@protect	86
add remark about Type 1 fonts required for automatic font expansion	8	\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning	85
Font sets: add OT4 encoding to text sets	128	\MT@maybe@do: redone	55
add T5 encoding to text sets	128	\MT@pdf@tex@no: compatibility with \TeX Live hack (reported by <i>Herbert Voß</i>)	38
declare qpl and qtm (qfonts, \TeX Gyre) as aliases of ppl resp. ptm	130	\MT@rem@from@clist: new macro: remove an item from a comma list	46
Inheritance: add list for OT4	133	\MT@scale@factor: generalised	61
add list for T5 (requested by <i>Hàn Thế Thành</i>)	134	\MT@toks: use instead of \toks@	39
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR	142	\SetProtrusion: (et al.) new key: font	101
settings for OT4 encoding (Computer Modern Roman, Palatino, Times)	137		

Version 1.9a (2005/12/05)

General: ‘(file name)/(line number)’ as default list name	104	diately (requested by <i>Georg Verwey</i>)	94
new option: defer@setup, by default true	114	\MT@get@highlevel: no longer check whether defaults have changed	95
remove superfluous test whether \pickup@font has changed	91	\MT@ifdefined@c@T: new macros: true case only	42
Documentation: add explanation for error message in DVI mode	27	\MT@ifint: use \pdfmatch if available	43
add explanation for error message with non-Type 1 fonts	27	\MT@ifstreq: use \pdfstrcmp if available	44
Font sets: declare mdbch (mathdesign) as an alias of Charter	130	\MT@in@clist: fix	46
Protrusion: fix: remove ‘_’ from OT1 encoding	142	\MT@info@missing@char: info instead of warning (after <i>Michael Hoppe</i> reported that the ‘fl’ ligature is missing in Palatino SC)	60
settings for T5 encoded Charter	137	\MT@is@feature: new macro: check for pdf \TeX feature	48
\microtypesetup: inside the preamble, accepts all package options	118	\MT@map@clist@n: following \LaTeX 3	45
\MT@check@font@cx: optimise context-sensitive setup	91	\MT@permute@@@@: don’t define permutations for unused encodings	110
\MT@define@set@key@: don’t expand variables immediately		\MT@rem@from@clist: fix	46
		\MT@setup@: defer setup until the end of the preamble	49

Version 1.9b (2006/01/20)

General: compatibility with listings: sanitise more catcodes (reported by <i>Holger Uhr</i>)	51	add samples of micro-typographic features	4
compatibility with the extendedchar option of the listings package	52	\MT@features: use throughout the package to adjust to beta-ness	48
Documentation: activate expansion in the distributed PDF	1	\MT@ifdimen: use \pdfmatch if available	44
		\MT@warn@code@too@large: fix calculation with present factor	61

Version 1.9c (2006/02/02)

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verwey</i>)	22	\MT@define@code@key@font: fix: context was ignored	104
Protrusion: settings for URW Garamond	139	\MT@define@code@key@size: fix: embrace \MT@tempsize in \csname (bug introduced in v1.9b)	103

Version 1.9d (2006/05/05)

Font sets: md* instead of m series in basic sets	128	\MT@get@font@dimen: warning for zero fontdimen . .	60
add QX encoding to text sets	128	\MT@get@opt: optimise: don't reset when preset op-	
Inheritance: add list for QX encoding (contributed by		tion is set	62
Maciej Eder)	133	set list name before presetting	62
Protrusion: settings for QX encoding (contributed by		\MT@is@active: support for Unicode (inputenc/utf8) .	86
Maciej Eder)	144	\MT@setupfont@hook: restore % and # when tex4ht	
settings for Euro symbols (Adobe, ITC, marvosym)	177	is loaded (reported by <i>Peter Dybala</i>)	50
tweak AMS settings	169	\SetProtrusion: (et al.) optimise: unify keys for	
\DeclareCharacterInheritance: fix: empty context	107	mandatory argument	101
\MT@detokenize@n: new macro: use \detokenize if		(et al.) split keys of optional and mandatory argu-	
available	43	ment	101
\MT@get@ex@opt: fix: evaluate preset	66		

Version 1.9e (2006/07/28)

General: fix: default value for activate: true . . .	113	settings for Euler Roman font	173
Documentation: add hint about unknown encodings	26	\DeclareCharacterInheritance: new key 'inputenc'	
include LPPL	200	to set the input encoding	107
Font sets: declare zeur and zeus (eulervm) as aliases		\MT@rem@from@cllist: model after \@removeelement	46
of eur resp. eus (euler)	130	\MT@setup@: empty \MT@setup@ after use (compatibil-	
Inheritance: adapt to marvosym's changed encoding	135	ity with the combine class)	49
Protrusion: complete settings for Euler Fraktur and		\pickup@font: no tracing with trace package	90
Script fonts	176	\SetExpansion: new key: inputenc	101
fix: forgotten comma in mt-mvs.cfg; adapt to		\SetProtrusion: (et al.) new key: inputenc	101
marvosym's changed encoding	177		

Version 1.9f (2006/09/09)

Protrusion: fix: euler-vm did not load euler settings	174	\MT@reset@context: only reset context if it has actu-	
\MT@curr@list@name: fix: \MessageBreak must not be		ally been changed	93
expanded	88	\MT@set@inh@list: fix: forgotten comma in the fea-	
\MT@gdef@n: new macros: global variants	41	tures list	108
\MT@get@inh@list: fix: input encoding must be set		\MT@set@named@keys: new macro: set name first, sim-	
after the inheritance list has been parsed	83	plify parsing of optional argument	103
\MT@glet: new macro	41	\SetProtrusion: (et al.) set catcodes before parsing	
		optional argument	101

Version 2.0 (2007/01/14)

General: (beta:1) new option: babel, by default false		Documentation: add hint about how to increase	
(language-dependent setup suggested by <i>Ulrich</i>		font_max and font_mem_size	28
<i>Dirr</i>)	114	add hint about warning when tracking and expan-	
(beta:1) new option: letterspace, by default 100	116	sion is applied to a font	28
(beta:3) compatibility with soul: register \textls		add remark about 'draft' option disabling	
and \lsstyle	52	microtype (noted by <i>Michalis Miatidis</i>)	9
(beta:8) option 'babel': fix: switch off French		qualify hint about web documents with regard to	
babel's shorthands properly (reported by <i>Daniel</i>		older pdfTeX versions	25
<i>Flipo</i>)	127	qualify hints about expansion error messages with	
(beta:8) option 'babel': switch off Turkish babel's		regard to older pdfTeX versions	27
shorthands	127	Font sets: add footnotesize and scriptsize sets	129
compatibility with listings: set catcode of back-		add smallcaps set	129
slash to zero (reported by <i>Steven Bath</i>)	52	\DeclareMicrotypeBabelHook: (beta:1) new com-	
maybe disable \MT@no ligatures after the preamble	126	mand: interaction with babel	101
new package letterspace: a stripped-down ver-		\lsstyle: (beta:1) new command: letterspacing . .	73
sion, containing the letterspacing commands		(beta:8) fix: font switches don't pose a problem	
only	1	anymore	73
option 'unit', \SetProtrusion: deprecate value		(beta:8) fix: letterspacing commands may be nes-	
'relative' completely	106	ted	73

totally redone, using the new <code>\letterspacefont</code>	73	<code>\SetExtraKerning</code> : (beta:1) new command: additional kerning	103
<code>\MT@declare@sets</code> : fix: empty size list when redefining set	94	<code>\SetExtraSpacing</code> : (beta:1) new command: adjustment of interword spacing	102
<code>\MT@is@symbol</code> : made even more robust	86	<code>\SetTracking</code> : new command: tracking	102
<code>\MT@load@inputenc</code> : sanitise catcodes before loading input encoding (problem with listings)	63	<code>\textls</code> : (beta:1) new command: letterspacing	77
<code>\MT@pdfTeX@no</code> : (beta:1) case 6: pdfTeX 1.40	37	starred version: remove spaces around text	77
<code>\MT@split@name</code> : adjust to possible letterspacing	55	<code>\tracingmicrotypeinpdf</code> : new debug method: mark all fonts with PDF annotations	36

Version 2.1 (2007/01/21)

General: compatibility with CJK: also check for its definition	89	<code>\lslig</code> : new command: protect ligatures in letterspaced text	73
compatibility with pinyin: disable microtype in <code>\py@macron</code> (reported by <i>Sven Naumann</i>)	52	<code>\MT@get@ls@basefont</code> : redone: use <code>\pdfmatch</code> to make it bullet-proof	73
fix: letterspace package forgot to load <code>keyval</code>	39	<code>\textls</code> : fix: use <code>\hmode@bgroup</code>	77

Version 2.2 (2007/07/14)

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i>)	49	fix: \TeX version shouldn't use <code>\x</code> and <code>\y</code> (found by <i>Wiebke Petersen</i>)	44
new option: <code>copyfonts</code>	115	<code>\MT@is@composite</code> : more robust: expand exactly once	87
simplify key declarations	104	<code>\MT@is@symbol</code> : expand once more (for frenchpro)	86
warning if stretch or shrink aren't multiples of step	123	<code>\MT@lsfont</code> : use <code>\font@name</code> , not <code>\MT@font</code>	70
Documentation: add hint about error message with pdfTeX 1.40	27	<code>\MT@maybe@etex</code> : use catcode trickery	41
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i>)	25	<code>\MT@pdfTeX@no</code> : case 7: pdfTeX 1.40.4	38
add overview	6	<code>\MT@preset@aux@space</code> : generalised	64
logo transparency and amusement	1	<code>\MT@requires@luatex</code> : (basic) support for luaTeX	41
Font sets: declare FPL Neu as an alias of Palatino	130	<code>\MT@set@all@pr</code> : (et al.) allow empty values	58
declare <code>chr</code> (<code>chmath</code>) as an alias of Charter (reported by <i>Geoff Vallis</i>)	130	<code>\MT@set@inputenc</code> : only load <code>inputenc</code> files if necessary	63
default set for tracking: <code>smallcaps</code>	129	<code>\MT@set@tr@codes</code> : disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	71
Inheritance: remove <code>'-'</code> \rightarrow <code>'127'</code>	132	possibility to customise interword spacing	70
Protrusion: settings for Bitstream Letter Gothic	138	<code>\MT@setupfont</code> : don't call <code>\@enc@update</code> anymore	53
Spacing: add sample	178	only add features that are available with the respective pdfTeX	53
Tracking: add ligatures that are to be disabled	135	<code>\MT@setupfont@hook</code> : restore percent character if Galician babel is loaded	50
<code>\DeclareMicrotypeVariants</code> : new command	99	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letterspaced fonts	58
<code>\DisableLigatures</code> : new optional argument: disable selected ligatures only	100	<code>\MT@tracking</code> : remember fonts that shouldn't be letterspaced	69
<code>\lslig</code> : always defined	73	<code>\MT@tracking@</code> : fix: tracking couldn't be re-enabled	69
<code>\MT@checklist@font</code> : fix: construct font name from characteristics	57	<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	125
<code>\MT@copy@font</code> : optionally work on copies of fonts	54	<code>\MT@with@babel@and@T</code> : also inspect class options	49
<code>\MT@get@basefamily</code> : redone, working on font names and suffixes of arbitrary length	81	<code>\pickup@font</code> : letterspace: setup inside group	90
<code>\MT@get@charwd</code> : subtract letterspacing amount from width	60	<code>\SetTracking</code> : new key 'no ligatures' to disable ligatures of letterspaced fonts	102
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	73	new keys 'spacing' and 'outer spacing' to adjust interword spacing (suggested by <i>Steven E. Harris</i>)	102
<code>\MT@ifdimen</code> : employ luaTeX features if available	44	third argument may be empty	102
<code>\MT@ifint</code> : employ luaTeX features if available	43	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code>	92
<code>\MT@ifstreq</code> : employ luaTeX features if available	44		

Version 2.3 (2007/12/23)

General: disable <code>\microtypecontext</code> in <code>hyperref</code> 's <code>\pdfstringdef</code>	51	<code>\MT@in@cllist</code> : fix: don't use <code>\x</code> (reported by <i>Peter Meier</i>)	46
enable protrusion when tracking is enabled . . .	124	<code>\MT@is@active</code> : support for extended Unicode (<code>inputenc/utf8x</code> resp. <code>ucs</code>) – experimental . . .	86
fix: really switch off Turkish shorthands	127	<code>\MT@noligatures</code> : fix: set evaluation didn't work (bug introduced in 2.2)	79
new value for verbose option: silent (suggested by <i>Karl Berry</i>)	115	<code>\MT@plain</code> : letterspace: support for <code>eplain/miniltx</code> . . .	37
turned some warnings into errors	115	<code>\MT@set@curr@os</code> : adjusting spaces made more reli- able	71
Documentation: add kerning sample	19	<code>\MT@set@tr@codes</code> : also adjust tracking if protrusion is not enabled, and even for letterspace (reported by <i>Stephan Hennig</i>)	71
add letterspacing illustration	17	possibility to customise outer kerning (suggested by <i>Stephan Hennig</i>)	71
<code>\do@subst@correction</code> : remember substitute font for all times (reported by <i>Stephan Hennig</i>)	91	<code>\MT@SetTracking</code> : sanity check for value	102
<code>\lslig</code> : redone: extract outer kerns from current letterspacing amount	73	<code>\MT@tr@outer@l</code> : only change pre outer space if it contains shrink	75
<code>\microtypecontext</code> : made robust (reported by <i>Ste- phan Hennig</i>)	92		
<code>\MT@begin@catcodes</code> : fix: don't disable <code>\KV@sp@def</code>	81		
<code>\MT@define@set@key@font</code> : font: single asterisk means normal font	97		

Version 2.3a (2008/02/29)

General: error messages if <code>pdfTeX</code> is too old for exten- sions	125	<code>\MT@fix@catcode</code> : fix catcodes earlier, and also for the letterspace package	34
fix test for <code>soul</code> under plain <code>TeX</code>	52	<code>\MT@getkey</code> : fix: <code>key=val</code> in class options list	121
Documentation: add hint about <code>babel</code> having to be loaded first	27	<code>\MT@maybe@etex</code> : removed	41
add table of available and enabled features	7	<code>\MT@set@codes</code> : generalised	59
mention <code>soulutf8</code>	29	<code>\MT@setupfont@hook</code> : restore percent character if Mexican <code>babel</code> is loaded	50
Protrusion: adjust LMR quotation marks again . . .	144		

Version 2.3b (2008/06/04)

General: compatibility with <code>CJKutf8</code> : also check for its definition	89	known slots	79
<code>\MT@exp@gcs</code> : new macro: reduce save stack size . .	41	<code>\MT@requires@latex</code> : new macro	37
<code>\MT@font@copy</code> : enable font copies also with protru- sion contexts (reported by <i>Nathan Rosenblum</i>)	54	<code>\MT@set@tr@codes</code> : fix: protrusion adjustment only for new fonts (reported by <i>Wolfram Schaalo</i>) . .	71
<code>\MT@get@size</code> : grouping	97	<code>\MT@tr@outer@l</code> : fix: only in horizontal mode	75
<code>\MT@noligatures@</code> : fix: warning messages for un- known slots	79	make <code>\spaceskip-aware</code> (<code>ragged2e</code>)	75
		<code>\MT@tr@outer@r@</code> : additional test for horizontal mode	76

Version 2.3c (2008/11/11)

General: <code>luaTeX</code> supported by default	39	coding (reported by <i>Vasile Gaburici</i>)	134
Documentation: add hint about partial incompatibil- ity with <code>CJK</code>	27	<code>\MT@detokenize@c</code> : fix: remove last space only (repor- ted by <i>Ulrich Durr</i>)	43
add hint about spacing being experimental	26	<code>\MT@tr@outer@r@</code> : additional test for horizontal mode (reported by <i>Sveinung Heggen</i>)	76
Inheritance: add <code>\textcommabelow[STst]</code> to <code>QX</code> en-			

Version 2.3d (2009/03/27)

General: default step: 1 for <code>pdfTeX</code> versions ≥ 1.40	122	<code>\microtypesetup</code> : select font after setup	118
fix pinyin compatibility check (reported by <i>Silas S. Brown</i>)	52	<code>\MT@check@active@set</code> : warning for missing default sets	117
move setup to the very end (for <i>Colin Rourke</i>) . .	127	<code>\MT@lua</code> : update for <code>luaTeX</code> 0.36	41
<code>\ifMT@inannot</code> : use <code>pdfTeXcmds</code> for debugging	36	<code>\MT@set@tr@codes</code> : allow zero tracking	70
<code>\lsstyle</code> : disable for <code>luaTeX</code>	73	<code>\MT@set@tr@zero</code> : fix: allow switching off tracking (reported by <i>Ulrich Durr</i>)	74
make invalid in math mode	73		

<code>\MT@tr@outer@r@</code> : don't use <code>\x</code> (reported by <i>Ulrich Dirr</i>)	76	<i>Maverick Woo</i>)	76
fix: don't adjust in math mode (reported by <i>Christoph Bier</i>)	76	<code>\MT@tr@set@okern</code> : allow empty value for outer kerning	78
fix: don't adjust inside discretionary (reported by		<code>\textls</code> : make math mode aware	77

Version 2.3e (2009/11/09)

General: compatibility fix for T _E X Live 2009 luaT _E X (reported by <i>Élie Roux</i>)	43	Spacing: settings for T2A encoding (contributed by <i>Karl Karlsson</i>)	180
Documentation: suggest to patch <code>\verbatim</code> instead of <code>\verbatim</code>	26	<code>\MT@get@font@dimen@six</code> : fix: gobbling settings with tracking failed (reported by <i>Leo</i>)	58
Expansion: settings for T2A encoding (contributed by <i>Karl Karlsson</i>)	136	<code>\MT@ls@aftergroup</code> : compatibility with <i>tikz</i> (first reported by <i>Christian Stark</i>)	72
Font sets: <code>sc*</code> instead of <code>sc</code> in <code>smallcaps</code> set	129	<code>\MT@setup@</code> : make space-unaware (requested by <i>Marcin Borkowski</i>)	49
add T2A encoding	128	<code>\MT@tr@outer@r@</code> : fix: set current kerning and spacing again (found by <i>Lars Rönnbäck</i>)	76
Protrusion: settings for T2A encoding (contributed by <i>Karl Karlsson</i>)	144		

B Index

Links (in blue) refer to the page where the corresponding entry is described (bold face) resp. occurs. Plain numbers (in black) refer to the code line where the corresponding entry is defined (underlined>) resp. used.

Options:		<code>\microtypesetup</code>	10
<code>DVIoutput</code>	9	<code>\SetExpansion</code>	14
<code>activate</code>	6	<code>\SetExtraKerning</code>	18
<code>auto</code>	8	<code>\SetExtraSpacing</code>	19
<code>babel</code>	9	<code>\SetProtrusion</code>	13
<code>config</code>	9	<code>\SetTracking</code>	16
<code>draft</code>	9	<code>\textls</code>	23
<code>expansion</code>	6	<code>\textls*</code>	23
<code>factor</code>	7	<code>\textmicrotypecontext</code>	22
<code>final</code>	9	<code>\UseMicrotypeSet</code>	11
<code>kerning</code>	7		
<code>letterspace</code>	9	A	
<code>protrusion</code>	6	<code>aoposter</code> (package)	97
<code>selected</code>	8	<code>activate</code> (option)	6, 113, 189
<code>shrink</code>	8	<code>\add@accent</code>	2386
<code>spacing</code>	7	<code>ae</code> (package)	21, 129
<code>step</code>	8	<code>amssymb</code> (package)	169
<code>stretch</code>	8	<code>article</code> (package)	22
<code>tracking</code>	7	<code>auto</code> (option)	8, 114
<code>unit</code>	8		
<code>verbose</code>	9	B	
User Commands:		<code>babel</code> (option)	9, 23, 27, 30, 114, 127, 189
<code>\DeclareCharacterInheritance</code>	20	<code>babel</code> (package)	3, 5, 17, 23, 24, 27, 50, 80, 101, 126, 127, 130, 186, 187, 189–191
<code>\DeclareMicrotypeAlias</code>	21		
<code>\DeclareMicrotypeBabelHook</code>	23	C	
<code>\DeclareMicrotypeSet</code>	10	<code>chemsym</code> (package)	186
<code>\DeclareMicrotypeSet*</code>	10	<code>chmath</code> (package)	21, 130, 190
<code>\DeclareMicrotypeSetDefault</code>	12	<code>CJK</code> (package)	27, 52, 89, 190, 191
<code>\DeclareMicrotypeVariants</code>	21	<code>CJKutef8</code> (package)	89, 191
<code>\DeclareMicrotypeVariants*</code>	21	<code>cm-super</code> (package)	8
<code>\DisableLigatures</code>	25	<code>color</code> (package)	9, 121
<code>\LoadMicrotypeFile</code>	22	<code>combine</code> (package)	127, 189
<code>\lslig</code>	24	<code>config</code> (option)	9, 20, 31, 117, 187
<code>\lststyle</code>	23	<code>contour</code> (package)	121
<code>\microtypecontext</code>	22	<code>copyfonts</code> (option)	54, 114, 115, 190

crop (package) 121
 csquotes (package) 28, 50, 142, 187
 \curr@fontshape 2333, 2335, 2337, 2344, 2384

D

\DeclareCharacterInheritance 20, 41, 3105
 \DeclareMicrotypeAlias 21, 43, 70, 2762
 \DeclareMicrotypeBabelHook 23, 45, 64, 2815
 \DeclareMicrotypeSet 10, 32, 57, 59, 71, 2493
 \DeclareMicrotypeSet* 10, 2493
 \DeclareMicrotypeSetDefault 12, 34, 61, 2721
 \DeclareMicrotypeVariants 21, 42, 58, 63, 2749
 \DeclareMicrotypeVariants* 21
 deferSetup (option) 50, 114, 118, 188
 \define@newfont 2326, 2334, 2338, 2346
 \DisableLigatures 25, 40, 62, 2795
 \do@subst@correction 2383
 docstrip (package) 33
 draft (option) 9, 114, 115, 189
 dsfont (package) 58
 DVIOutput (option) 9, 114, 186

E

eco (package) 21, 129
 \efcode 1304, 1328, 1329, 1369, 1371
 eplain (package) 24, 30, 33, 37, 49, 120, 191
 eucal (package) 174
 eufrak (package) 176
 euler (package) 88, 173, 189
 eulervm (package) 21, 130, 174, 189
 euroitc (package) 177
 europms (package) 177
 eurosans (package) 177
 expansion (option) 6, 7, 11, 113, 121

F

\f@family 2340, 2341
 \f@size 2333, 2335, 2337, 2344, 2384
 factor (option) 7, 13, 31, 116, 186
 fancyvrb (package) 26
 final (option) 9, 32, 114, 115, 186
 fix-cm (package) 55
 \font 807, 1670
 \font@name ... 110, 843, 845, 847, 849, 859, 860,
 1554, 1559, 1593, 1599, 1608, 1611, 1657,
 1670, 1682, 1684, 1689, 1691, 1694, 1697,
 1700, 1732, 1787, 1905, 1907, 2326, 2334,
 2343, 2346, 2367, 2369, 2371, 2385, 2397, 2417
 \fontcharwd 1068
 \fontdimen 996,
 1003, 1089, 1094, 1193, 1196, 1571, 1572,
 1599, 1682, 1725, 1726, 1733, 1734, 1740,
 1744, 1751, 1761, 1765, 1768, 1787, 1918, 1921
 fontinst (package) 181
 fontinstallationguide (package) 159
 fourier (package) 58
 french (package) 187
 frenchpro (package) 86, 186, 190

G

german (package) 34, 185
 graphics (package) 9, 121

H

hfoldsty (package) 21, 129
 hyperref (package) . 9, 50, 51, 89, 121, 187, 188, 191

I

IEEEtran (package) 89
 \iffontchar 1099
 \ifMT@auto 240, 3770, 3817
 \ifMT@babel 240, 3971, 4014
 \ifMT@do 879, 942, 1531, 1937
 \ifMT@document 282, 3221
 \ifMT@draft 240, 3706
 \ifMT@expansion 240, 3737, 3814
 \ifMT@if@ 239,
 710, 716, 734, 765, 2411, 3324, 3991, 3994
 \ifMT@inannot 98
 \ifMT@inlist@ 540, 585, 644,
 667, 822, 909, 926, 935, 953, 971, 1529,
 1988, 1999, 2048, 2096, 2405, 2467, 2479, 2780
 \ifMT@kerning 240, 3881, 3930, 3984
 \ifMT@noligatures 240, 3940
 \ifMT@nonselected 1282, 2109, 2117
 \ifMT@norest 2171, 2178
 \ifMT@opt@auto 3331, 3773
 \ifMT@opt@DVI 3331, 3722
 \ifMT@opt@expansion 3331, 3717
 \ifMT@protrusion 240, 673, 695, 3725, 3864
 \ifMT@selected 240, 3800, 3820
 \ifMT@spacing 240, 3870, 3924, 4013
 \ifMT@tracking 240, 3859, 3918
 ifpdf (package) 9, 89, 121
 \iftracingmicrotypeinpdfall 104
 inputenc (package)
 .. 14, 27, 31, 52, 63, 84, 86, 88, 187, 189–191

J

jurabib (package) 89

K

kerning (option) 7, 11, 30, 113
 keyval (package) 39, 59, 131, 190
 \knacode 1484, 1485, 1495, 1498, 1504
 \knbccode 1477, 1478, 1494, 1497, 1503
 \knbscode 1398, 1399, 1422, 1426, 1432

L

ledmac (package) 29, 31, 49, 89, 187
 ledpar (package) 29
 \leftmarginkern 681
 letterspace (option) . 9, 18, 24, 30, 43, 70, 116, 189
 letterspace (package)
 1, 24, 30, 33, 37, 49, 70–73, 189–191
 \letterspacefont 1559
 lineno (package) 49
 listings (package) 26, 51, 52, 63, 80, 188–190
 lmodern (package) 8, 144
 \LoadMicrotypeFile 22, 44, 73, 2776, 3711
 \lpcode 1010, 1042, 1043, 1229, 1232, 1584
 \lslig 24, 54, 1675, 1678
 \lststyle 23, 51, 757, 780, 785,
 1658, 1848, 1851, 1856, 1859, 1870, 1874, 3714

M

- marvosym (package) 21, 31, 33, 135, 177, 189
- mathdesign (package) 21, 130, 188
- \mbox 129
- memoir (package) 23, 89
- \Microtype@Hook 117, 3563
- \microtypecontext
 - . . . 22, 47, 755, 2437, 2447, 3713, 3954, 3957
- microtypecontext (environment) 22
- \microtypesetup 10, 46, 3567, 3712
- miniltx (package) 24, 30, 33, 37, 49, 191
- minimal (package) 52
- \MT@font 331, 825,
 - 841, 998, 1090, 1103, 1542, 2051, 2122, 2372
- \MT@abbr@ex 620
- \MT@abbr@ex@c 620
- \MT@abbr@ex@inh 620
- \MT@abbr@kn 620
- \MT@abbr@kn@c 620
- \MT@abbr@kn@inh 620
- \MT@abbr@nl 620
- \MT@abbr@pr 620
- \MT@abbr@pr@c 620
- \MT@abbr@pr@inh 620
- \MT@abbr@sp 620
- \MT@abbr@sp@c 620
- \MT@abbr@sp@inh 620
- \MT@abbr@tr 620
- \MT@abbr@tr@c 620
- \MT@active@features
 - 851, 2399, 2402, 2414, 2421, 2456, 2466,
 - 2802, 3726, 3815, 3861, 3872, 3883, 3946, 3947
- \MT@addto@annot 98
- \MT@addto@setup 659, 660, 719, 1206,
 - 1623, 2320, 2437, 2438, 3000, 3018, 3435,
 - 3568, 3705, 3857, 3917, 3939, 3945, 3970, 4003
- \MT@auto 1291, 3769, 3783, 3793, 3797
- \MT@auto@ 1291, 1300, 1342
- \MT@autofalse 242, 3781, 3792
- \MT@autotrue 242, 3525
- \MT@babelfalse 249
- \MT@babeltrue 249
- \MT@begin@catcodes 1989, 1990, 2030, 2784, 2797,
 - 2822, 2836, 2858, 2877, 2891, 3110, 3538, 3539
- \MT@cat 1202, 2312, 2314
- \MT@cfg@catcodes 767, 1224, 2015, 2032
- \MT@char 1032, 1042, 1043, 1048, 1049, 1052, 1054,
 - 1068, 1069, 1099, 1229–1233, 1328, 1329,
 - 1331, 1332, 1369–1371, 1398, 1399, 1405,
 - 1406, 1412, 1413, 1416, 1417, 1422–1427,
 - 1477, 1478, 1484, 1485, 1488, 1489, 1494–
 - 1498, 1948, 2151, 2167, 2168, 2173, 2244,
 - 2247, 2249, 2275, 2278, 3170, 3171, 3176, 3177
- \MT@char@
 - 2151, 2157, 2162, 2167, 2181, 2183, 2189,
 - 2190, 2192, 2205, 2206, 2209, 2210, 2213,
 - 2214, 2218, 2220, 2248, 2261, 2264, 2267, 2286
- \MT@charstring 2164, 2247, 2252
- \MT@check@active@set
 - 3549, 3732, 3833, 3863, 3875, 3887
- \MT@check@font 821, 2396, 2458
- \MT@check@font@cx 2400, 2458
- \MT@check@range 3295, 3297
- \MT@check@range@ 3297, 3298
- \MT@check@rlist 3248, 3288
- \MT@check@rlist@ 3288, 3289
- \MT@check@step 3821
- \MT@checklist@ 887, 902, 1934
- \MT@checklist@family 920
- \MT@checklist@font 964
- \MT@checklist@size 948
- \MT@checksetup 3577, 3588, 3595, 3614, 3634
- \MT@clear@options 192, 216, 231, 655, 3699
- \MT@clist@break 512, 915, 943, 959, 973
- \MT@clist@function 512
- \MT@cnt@encoding 3187, 3195, 3196
- \MT@cnt@family 3193, 3202, 3203
- \MT@cnt@series 3200, 3209, 3210
- \MT@cnt@shape 3207, 3215, 3216
- \MT@config@file 3528, 3536, 3537, 3541, 3542, 3545
- \MT@context 2072, 2084, 2103, 2130
- \MT@copy@font 810,
 - 838, 2996, 3002, 3014, 3021, 3452, 3454, 3723
- \MT@copy@font@ 838, 2996, 3002, 3014, 3021, 3452
- \MT@count 450, 451, 604, 1061, 1068, 1070,
 - 1071, 1076, 1081, 1082, 1086, 1094, 1125, 1187
- \MT@curr@file 1993, 1994,
 - 2003, 2004, 2786, 2787, 2969, 3132, 3541, 4005
- \MT@curr@list@name 1092, 1128, 1218, 1338,
 - 1945, 1974, 2141, 2281, 2289, 2294, 2302, 2308
- \MT@curr@ls 1590, 1679, 1682
- \MT@curr@set@name
 - 2518, 2520–2523, 2525, 2526, 2531, 2536,
 - 2540, 2541, 2566, 2570, 2599, 2629, 2635,
 - 2641, 2645, 2646, 2873, 2931, 2949, 2964,
 - 2976, 3040, 3044, 3049, 3052, 3054, 3058,
 - 3061, 3064, 3074, 3087, 3093, 3101, 3133, 3135
- \MT@declare@char@inh 3115, 3120, 3130
- \MT@declare@sets 2501, 2507, 2519, 2804
- \MT@DeclareSet 2496, 2498, 2515
- \MT@DeclareSetAndUseIt 2495, 2514
- \MT@DeclareVariants 2752, 2753, 2755
- \MT@def@bool@opt 3406,
 - 3419–3421, 3430, 3442, 3450, 3459, 3466
- \MT@def@n 320, 2489, 2490
- \MT@default@ex@set 2737
- \MT@default@kn@set 2737
- \MT@default@pr@set 2737
- \MT@default@sp@set 2737
- \MT@default@tr@set 2737
- \MT@define@code@key 2913, 2956–2959, 3155–3157
- \MT@define@code@key@font 2936, 2961, 3159
- \MT@define@code@key@size 2924, 2960, 3158
- \MT@define@opt@key 2962, 2978–2981, 3097–3099
- \MT@define@optionX 3567, 3623, 3624
- \MT@define@optionX@ 3625, 3653, 3655, 3656
- \MT@define@set@key@ 2529, 2685–2688
- \MT@define@set@key@font 2633, 2690
- \MT@define@set@key@size 2559, 2689
- \MT@detokenize@c 361, 2159, 2245
- \MT@detokenize@n 361, 2277
- \MT@dimen@six 995, 1062, 1086, 1125, 1760, 1763, 1914
- \MT@dinfo 86
- \MT@dinfo@list 900, 910, 913, 918, 927,
 - 930, 936, 938, 946, 954, 957, 962, 972, 975, 979
- \MT@dinfo@nl 86

- \MT@DisableLigatures 2795
 \MT@do@font 602, 1012, 1304, 1435, 1505, 1584
 \MT@documentfalse 282
 \MT@documenttrue 282, 3949
 \MT@dofalse 879, 891, 914, 931, 958, 976
 \MT@dotrue 879, 882, 911, 928, 937, 955, 1929
 \MT@draftfalse 245, 3468
 \MT@drafttrue 245, 3470
 \MT@edef@n 322, 2108, 2134,
 2483, 2919, 2969, 2971, 3131, 3146, 3275, 3499
 \MT@encoding 869, 968,
 2067, 2079, 2159, 2245, 2276, 2293, 2300, 2308
 \MT@encoding@check 3144, 3148
 \MT@end@catcodes 1991, 2034, 2792,
 2806, 2833, 2855, 2874, 2888, 2902, 3125, 3540
 \MT@error 75, 221, 647, 722, 1849,
 1857, 1966, 1979, 2597, 2716, 2743, 2810,
 2988, 3005, 3030, 3444, 3461, 3600, 3618,
 3642, 3774, 3786, 3850, 3919, 3925, 3931, 3966
 \MT@ex@cc@name 1275,
 1277, 1344, 1345, 1355, 1361, 1362, 1374, 2839
 \MT@ex@context 841, 2450, 2488
 \MT@ex@doc@contexts 2488
 \MT@ex@factor 250, 1292, 1348
 \MT@ex@factor@ . 1292, 1306, 1309, 1318, 1319, 1342
 \MT@ex@inh@name 1330–1332
 \MT@ex@level 250, 3816, 3818
 \MT@ex@max 262, 1321, 1322
 \MT@ex@min 262, 1324, 1325
 \MT@ex@setname 2708
 \MT@ex@split@val 1316
 \MT@exp@cs 317, 320, 323,
 326, 329, 852, 854, 863, 952, 1053, 1112,
 1115, 1145, 1149, 1159, 1162, 1165, 1332,
 1417, 1489, 2095, 2415, 2422, 2423, 2427,
 2469, 2480, 2565, 2930, 2947, 3177, 3250, 3293
 \MT@exp@gcs 317, 321, 325, 327, 330
 \MT@exp@one@n 332,
 366, 519, 643, 813, 815, 864, 906, 969,
 1528, 1998, 2396, 2403, 2433, 2466, 2477,
 2488, 2507, 2684, 2701, 2730, 2779, 3120, 3954
 \MT@exp@two@c 334,
 352, 358, 367, 554, 812, 845, 847, 849,
 858, 1657, 1691, 1694, 1697, 2158, 2163, 2246
 \MT@exp@two@n 336, 924, 933, 2535, 2640
 \MT@expandfont 1279, 1295, 1299
 \MT@expansion 827, 1267, 3843
 \MT@expansionfalse 241, 3718, 3810
 \MT@expansiontrue 241, 3524
 \MT@extra@context 2488, 2825, 2839,
 2862, 2880, 2894, 2942, 2945, 2946, 2948,
 2982, 2997, 3003, 3015, 3020, 3106, 3247,
 3251, 3254, 3257, 3260, 3264, 3265, 3267, 3294
 \MT@extra@inputenc 3105, 3129, 3134, 3135
 \MT@factor@default 274, 3505, 3729
 \MT@family ... 813, 869, 925, 968, 2068, 2771, 2772
 \MT@familyalias
 .. 814, 815, 875, 932, 934, 2078, 2080, 2773
 \MT@feat 879, 1000, 1034, 1107, 1109,
 1111, 1112, 1114, 1115, 1121, 1123, 1126,
 1134–1138, 1140, 1142–1146, 1149, 1150,
 1155, 1159, 1162, 1165, 1168–1170, 1179,
 1180, 1185, 1203, 1660, 1964, 1966, 1969,
 1970, 1974, 1976, 1979, 2104, 2106, 2108,
 2112, 2116, 2120, 2121, 2124, 2132, 2134,
 2135, 2137, 2144, 2148, 2282, 2283, 2312, 2314
 \MT@features
 640, 2488, 2501, 2684, 2695, 2724, 2966, 3115
 \MT@features@long 640, 643, 648, 2463, 3127
 \MT@file@list 1985,
 1987, 1995, 1998, 2000, 2005, 2008, 2779, 2783
 \MT@find@file 813, 815, 1985
 \MT@fix@catcode 5
 \MT@fix@font@set 2554, 3724
 \MT@font 331,
 812, 858, 865, 996, 1003, 1010, 1011, 1042,
 1043, 1048, 1049, 1068, 1069, 1089, 1094,
 1099, 1193, 1196, 1229, 1230, 1232, 1233,
 1300, 1304, 1328, 1329, 1369, 1371, 1398,
 1399, 1405, 1406, 1412, 1413, 1422–1424,
 1426, 1427, 1432–1434, 1477, 1478, 1484,
 1485, 1494, 1495, 1497, 1498, 1503, 1504,
 1528, 1532, 1589, 1938, 2316, 2367–2369,
 2371, 2384, 2385, 2396, 2398, 2403, 2425, 2434
 \MT@font@copy 841, 842, 849, 850, 858–860
 \MT@font@list 2316, 2396–2398, 2457
 \MT@font@orig 842
 \MT@font@sets 2539, 2554, 2644, 3724
 \MT@gdef@n 320, 2818, 2832,
 2854, 2887, 2901, 3054, 3074, 3087, 3137, 3557
 \MT@get@axis 2658–2661, 2671
 \MT@get@basefamily 1997, 2035
 \MT@get@basefamily@ 2041, 2044
 \MT@get@char@unit 1033, 1157, 1183, 1271
 \MT@get@charwd 1067, 1086, 1160, 1183
 \MT@get@config 3528
 \MT@get@ex@opt 1270, 1286, 1342
 \MT@get@ex@opt@ 1350–1352, 1354, 1360
 \MT@get@font 2639, 2649
 \MT@get@font@ 2650, 2655, 2954
 \MT@get@font@and@size 2941, 2953
 \MT@get@font@dimen 1088, 1163
 \MT@get@font@dimen@six . 984, 995, 1382, 1461, 1543
 \MT@get@highlevel ... 2534, 2544, 2673, 2918, 3145
 \MT@get@inh@list 987, 1273, 1385, 1464, 2130
 \MT@get@listname 2050, 2106, 2132
 \MT@get@listname@ 2050
 \MT@get@ls@basefont 1558, 1681, 1688
 \MT@get@opt 985, 1132, 1383, 1462
 \MT@get@range 2563, 2573, 2928
 \MT@get@size 2580, 2585, 2594, 2613, 2669
 \MT@get@slot 1031, 1947, 2151, 3169, 3175
 \MT@get@space@unit
 .. 1157, 1257, 1396, 1403, 1410, 1475, 1482
 \MT@get@tr@opt 1545, 1629
 \MT@get@tr@opt@ 1646–1649, 1651
 \MT@get@unit 1165, 1173, 1639
 \MT@get@unit@ 1173
 \MT@getkey 3686, 3701
 \MT@glet 316, 327, 845, 1691,
 1720, 1851, 1859, 2365, 2458–2461, 2773,
 2996, 3014, 3190, 3452, 3454, 3901, 3906, 4008
 \MT@glet@nc 326, 1732, 1905, 2144,
 2385, 2457, 2523, 2525, 2531, 2635, 2847,
 3040, 3044, 3049, 3052, 3058, 3061, 3064, 3093
 \MT@glet@nn 329, 2476, 3554

- `\MT@if@false` 239,
 706, 713, 730, 747, 2408, 3292, 3985, 3992
`\MT@if@list@exists`
 ... 983, 1269, 1285, 1381, 1460, 1544, 2103
`\MT@if@true`
 239, 707–709, 714, 715, 731–733, 760, 764,
 2401, 3301, 3305, 3313, 3318, 3986–3990, 3993
`\MT@ifdefined@c@T`
 .. 340, 710, 736, 1051, 1075, 1330, 1415,
 1487, 1577, 2233, 2771, 3134, 3237, 3563, 3566
`\MT@ifdefined@c@TF`
 .. 340, 388, 674, 1548, 1561, 1564, 1575,
 1679, 1757, 1911, 1942, 1943, 2107, 2133, 3246
`\MT@ifdefined@n@T`
 340, 664, 903, 921, 949, 965, 1052,
 1168, 1204, 1331, 1355, 1416, 1488, 1631,
 1633, 1652, 2086, 2311, 2521, 2766, 2842,
 2972, 3196, 3203, 3210, 3216, 3247, 3257, 3686
`\MT@ifdefined@n@TF` 340, 875,
 885, 904, 922, 950, 966, 1134, 1142, 1344,
 1361, 1932, 1969, 2074, 2159, 2710, 2713,
 2739, 3223, 3274, 3283, 3347, 3550, 3553, 3952
`\MT@ifdim` .. 459, 574, 575, 579, 580, 2596, 2605,
 3299–3301, 3303, 3304, 3311–3313, 3316, 3317
`\MT@ifdimen` 424, 2624, 3043, 3051, 3063, 3514
`\MT@ifempty` . 373, 1010, 1011, 1040, 1046, 1246,
 1247, 1395, 1402, 1409, 1432–1434, 1446–
 1448, 1450–1452, 1474, 1481, 1503, 1504,
 1520, 1521, 1738, 1739, 1749, 1767, 1882,
 1903, 1904, 1916, 2468, 2500, 2505, 2546,
 2552, 2574, 2575, 2588, 2589, 2663, 2674,
 2694, 2699, 2723, 2728, 2801, 2869, 2963,
 2968, 2982, 2995, 3003, 3013, 3020, 3072,
 3114, 3119, 3149, 3360, 3391, 3530, 3574, 3631
`\MT@ifint` 392, 2212, 2870, 3073, 3498, 3507
`\MT@ifstreq` 466, 1210, 1965, 2105, 2120,
 2262, 2265, 2432, 2638, 2940, 3042, 3060,
 3062, 3238, 3363, 3365, 3368, 3371, 3393,
 3395, 3409, 3410, 3476, 3477, 3482, 3486,
 3513, 3576, 3583, 3587, 3594, 3633, 3639, 4016
`\MT@in@clist` 540, 643, 665, 906, 924, 933,
 969, 1528, 1987, 1998, 2396, 2403, 2466, 2779
`\MT@in@rlist` 568, 952, 2095
`\MT@in@rlist@` 568
`\MT@in@rlist@@` 568
`\MT@in@tlist` 556, 2047, 2477
`\MT@in@tlist@` 556
`\MT@inannotfalse` 99
`\MT@inannottrue` 99
`\MT@increment` 604, 3195, 3202, 3209, 3215
`\MT@info` 75, 90, 95, 3483, 3671, 3675
`\MT@info@missing@char` 1071, 1097
`\MT@info@nl` 75, 91, 92, 96, 675, 1098,
 3474, 3484, 3537, 3551, 3555, 3721, 3723,
 3728, 3735, 3817, 3844, 3862, 3868, 3874,
 3879, 3886, 3891, 3922, 3928, 3934, 3962, 3973
`\MT@inh@do` 2143, 3160
`\MT@inh@feat` 3105, 3114, 3117, 3128
`\MT@inh@split` 3162, 3166
`\MT@inlist@false` 540, 544, 557, 569, 2411
`\MT@inlist@true` 540, 546, 564, 576, 581, 2411
`\MT@is@active` 2158, 2225
`\MT@is@char` 2163, 2246, 2252
`\MT@is@composite` 2161, 2273
`\MT@is@feature` 642, 2506, 2700, 2729
`\MT@is@letter` 2156, 2179, 2249, 2278
`\MT@is@number` 2195, 2200
`\MT@is@symbol` 2160, 2243
`\MT@iterate` 590
`\MT@kerning` 830, 1457, 3890
`\MT@kerningfalse` 247
`\MT@kerningtrue` 247
`\MT@kn@c@name` 1466, 1468, 1511, 2894
`\MT@kn@context` 2451, 2488
`\MT@kn@doc@contexts` 2488
`\MT@kn@factor` 250
`\MT@kn@factor@` 1134
`\MT@kn@inh@name` 1487–1489
`\MT@kn@max` 262
`\MT@kn@min` 262
`\MT@kn@setname` 2708
`\MT@kn@split@val` 1472
`\MT@kn@unit` 256
`\MT@kn@unit@` 1142, 1514
`\MT@led@kern` 671
`\MT@led@unhbox@line` 671
`\MT@led@mac@setup` 671, 728
`\MT@let@cn` 328, 329, 330, 876, 991, 1277,
 1345, 1389, 1468, 1632, 1634, 1638, 1964,
 1976, 2100, 2104, 2137, 2367, 2617, 2976, 3133
`\MT@let@nc` 326, 670, 1170, 2116, 2148,
 2406, 3276, 3281, 3369, 3372, 3590, 3597, 3853
`\MT@let@nn`
 329, 1135, 1140, 1143, 1155, 1362, 1365, 1653
`\MT@letterspace`
 ... 278, 1548, 1632, 1642, 3909, 3910, 3912
`\MT@letterspace@`
 1019, 1075, 1076, 1548, 1549, 1552,
 1555, 1559, 1563, 1571, 1583–1585, 1590,
 1599, 1763, 1767, 1868, 1883–1885, 1917, 1920
`\MT@letterspace@default` 278, 3910
`\MT@listname`
 2050, 2097, 2100, 2107, 2108, 2110, 2112,
 2133, 2134, 2136, 2137, 2141, 2144, 3177, 3182
`\MT@load@inputenc` 1211, 1214, 1223
`\MT@load@list` 989, 1275, 1387, 1466, 1962
`\MT@loop` 590, 598, 3280
`\MT@lower` 2567, 2573, 2931
`\MT@ls@adjust` 1597, 1879
`\MT@ls@adjust@` 1864, 1886
`\MT@ls@adjust@empty` 1865, 1879
`\MT@ls@adjust@relax` 1866, 1879
`\MT@ls@aftergroup` 1603, 1616, 1621
`\MT@ls@basefont` 1688, 1697, 1700, 1701
`\MT@ls@outer@k`
 .. 1600, 1609, 1621, 1780, 1794, 1835, 1925
`\MT@ls@set@ls` 1870, 1873, 1879
`\MT@ls@too@large` 1885, 1888, 3912
`\MT@ls@font` 1554, 1559,
 1571, 1572, 1581, 1584, 1585, 1589, 1657,
 1666, 1710, 1712, 1725, 1726, 1733, 1734,
 1740, 1744, 1751, 1761, 1765, 1768, 1918, 1921
`\MT@lslig` 1678
`\MT@lua` 307, 396, 428, 470
`\MT@map@clist@` 512

- `\MT@map@clist@c` 512, 851, 1946, 2402,
 2414, 2421, 2456, 2463, 2501, 2503, 2695,
 2697, 2724, 2726, 2966, 3115, 3117, 3127, 3684
`\MT@map@clist@n` 512, 884,
 1931, 2488, 2532, 2561, 2636, 2684, 2756,
 2816, 2916, 2926, 2938, 3056, 3070, 3172,
 3355, 3358, 3387, 3389, 3418, 3493, 3572, 3629
`\MT@map@tlist@` 530
`\MT@map@tlist@c` 530, 559, 570, 854, 1053,
 1332, 1417, 1489, 2054, 2427, 3248, 3293, 3724
`\MT@map@tlist@n` 530, 2026, 2254, 3189
`\MT@maybe@do` 879, 981, 1267, 1379, 1458, 1530
`\MT@maybe@etex` 283
`\MT@maybe@rem@from@list` 2427, 2431
`\MT@MT` 2, 75, 78, 81, 83, 85,
 90, 91, 93, 94, 204, 212, 223, 654, 723, 725,
 3005, 3531, 3671, 3675, 3722, 3966, 3967, 4020
`\MT@next@listname` 2057, 2066
`\MT@next@listname@` 2075, 2087, 2093
`\MT@n@ligatures` 1938, 2795
`\MT@n@setname` 1930, 2795
`\MT@noligatures` 832, 1927, 3941
`\MT@noligatures@` 1710, 1712, 1938, 1941
`\MT@noligatures@false` 244
`\MT@noligatures@true` 244, 2803
`\MT@nonselected@false` 1282, 1296
`\MT@nonselected@true` 1282, 1284
`\MT@noreset@false` 2178, 2185, 2194, 2215, 2262, 2265
`\MT@noreset@true` 2154, 2178, 2203
`\MT@old@cmd` 65, 70–73
`\MT@opt@autofalse` 3332
`\MT@opt@autot@true` 3332, 3420
`\MT@opt@def@set` 3346, 3374, 3399
`\MT@opt@DVIfalse` 3333
`\MT@opt@DVIftrue` 3333, 3423, 3426
`\MT@opt@expansion@false` 3331
`\MT@opt@expansion@true` 3331
`\MT@options` 2904
`\MT@optwarn@admissible` 3334, 3411, 3486
`\MT@optwarn@nan` 3338, 3500, 3509
`\MT@orig@add@accent` 2386
`\MT@orig@foreign@language` 3979, 3981
`\MT@orig@pickupfont` 753, 794, 799, 2321, 2388
`\MT@orig@py@macron` 791, 795, 800
`\MT@orig@select@language` 3974, 3976
`\MT@outer@kern` ... 1599, 1602, 1607, 1609, 1610,
 1615, 1682, 1683, 1685, 1835, 1836, 1899, 1925
`\MT@outer@space`
 1593, 1594, 1723, 1776, 1778, 1779,
 1792, 1793, 1807, 1808, 1823, 1824, 1828, 1829
`\MT@pdf@annot` 98
`\MT@pdf@tex@no` 148, 198, 202, 296
`\MT@permute` 2831, 2853, 2867, 2886, 2900, 3140, 3186
`\MT@permute@` 3186
`\MT@permute@@` 3186
`\MT@permute@@@` 3186
`\MT@permute@@@@` 3214, 3219
`\MT@permute@@@@@` 3227, 3229
`\MT@permute@define` 3220, 3230–3232, 3272
`\MT@permute@reset` 3189, 3278
`\MT@permute@list`
 2825, 2839, 2865, 2880, 2894, 2942,
 2943, 2948, 3138, 3242, 3243, 3247, 3251,
 3254, 3257–3260, 3267, 3268, 3294, 3325, 3326
`\MT@pickupfont` 796, 801, 2382, 2390
`\MT@plain` 124, 145
`\MT@pr@c@name` 989, 991, 1237, 2825
`\MT@pr@c@context` 841, 2450, 2488
`\MT@pr@doc@contexts` 2488
`\MT@pr@factor` 250, 3508, 3729, 3730
`\MT@pr@factor@` 1134
`\MT@pr@inh@name` 1051, 1052, 1054
`\MT@pr@level` 250, 3727, 3728
`\MT@pr@max` 262
`\MT@pr@min` 262
`\MT@pr@setname` 2708
`\MT@pr@split@val` 1038
`\MT@pr@unit` 256, 3515, 3731
`\MT@pr@unit@` 1142, 1240
`\MT@preset@aux` 1242,
 1244, 1246, 1247, 1250, 1516, 1518, 1520, 1521
`\MT@preset@aux@factor` 1242, 1250, 1446–1448, 1516
`\MT@preset@aux@space` . 1244, 1250, 1450–1452, 1518
`\MT@preset@ex` 1356, 1373
`\MT@preset@kn` 1509
`\MT@preset@kn@` 1509
`\MT@preset@pr` 1235
`\MT@preset@pr@` 1235
`\MT@preset@sp` 1439
`\MT@preset@sp@` 1439
`\MT@ProcessOptionsWithKV` 3679, 3702
`\MT@protrusion` 826, 981, 3734
`\MT@protrusion@false` 240
`\MT@protrusion@true` 240, 3521
`\MT@rbba@expansion` 635
`\MT@rbba@kerning` 635
`\MT@rbba@protrusion` 635
`\MT@rbba@spacing` 635
`\MT@rbba@tracking` 635
`\MT@register@font` 835, 2398, 2459
`\MT@register@font@cx` 2420, 2459
`\MT@register@subst@font` 2373, 2397, 2460
`\MT@register@subst@font@cx` 2413, 2460
`\MT@rem@from@clist` 551, 864, 2433, 2908
`\MT@rem@from@list` 854, 862
`\MT@rem@last@space` 361
`\MT@repeat` 590, 600, 3286
`\MT@requires@latex` 144, 193, 657, 749,
 778, 1656, 2319, 3339, 3492, 3679, 3683, 4007
`\MT@requires@luatex` .. 299, 307, 387, 394, 426,
 468, 1855, 2986, 3441, 3626, 3860, 3871, 3882
`\MT@requires@pdftex` 295,
 392, 424, 466, 671, 809, 818, 829, 831, 839,
 1017, 1059, 1073, 1307, 1378, 1457, 1525,
 1707, 1927, 2795, 2984, 3086, 3092, 3439,
 3523, 3625, 3744, 3771, 3796, 3847, 3938, 4011
`\MT@res@a` ... 489, 491, 542, 549, 552, 554, 558, 563
`\MT@res@b` 490, 491, 552–554, 562, 563
`\MT@reset@context` 2441, 2444, 2448, 2475
`\MT@reset@context@` 2448, 2475
`\MT@reset@ef@codes` 1272, 1294, 1307, 1357
`\MT@reset@ef@codes@` 1302, 1310, 1314
`\MT@reset@kn@codes` 1463, 1470, 1500
`\MT@reset@kn@codes@` 1500
`\MT@reset@pr@codes` 986, 993, 1014

- \MT@reset@pr@codes@ 1014
 \MT@reset@sp@codes 1384, 1391, 1429
 \MT@reset@sp@codes@ 1429
 \MT@restore@cat@codes ... 5, 7, 8, 216, 231, 655, 4029
 \MT@restore@p@h 704, 716, 765
 \MT@saved@setupfont 3669, 3676
 \MT@scale 611, 1062, 1108, 1122, 1125, 1319
 \MT@scale@factor 1064, 1083, 1106, 1252, 1375
 \MT@scale@to@em .. 1041, 1047, 1059, 1080, 1258,
 1397, 1404, 1411, 1476, 1483, 1641, 1758, 1912
 \MT@selected@false 243
 \MT@selected@true 243
 \MT@series 869, 968, 2069, 2081
 \MT@set@all@ex 1302, 1376
 \MT@set@all@kn 1500, 1522
 \MT@set@all@pr 1007, 1014, 1248
 \MT@set@all@sp 1429, 1454
 \MT@set@babel@context 3951, 3977, 3982, 3996
 \MT@set@codes .. 992, 1022, 1278, 1390, 1469, 1977
 \MT@set@curr@ls 1590
 \MT@set@curr@ok
 .. 1602, 1615, 1621, 1627, 1628, 1812, 1816
 \MT@set@curr@os 1592, 1627, 1628, 1812, 1816
 \MT@set@default@set 2724, 2730, 2737
 \MT@set@ex@codes 1298, 3801, 3803
 \MT@set@ex@codes@n 1282, 1298, 3803
 \MT@set@ex@codes@os 1268, 3801
 \MT@set@ex@heirs 1332, 1368
 \MT@set@inh@list 3111, 3113
 \MT@set@input@enc 988, 1201, 1274, 1386, 1465, 2142
 \MT@set@input@enc@ 1204, 1206
 \MT@set@kn@codes 1459
 \MT@set@kn@heirs 1489, 1493
 \MT@set@list@name
 990, 1133, 1276, 1343, 1388, 1467, 1630, 2281
 \MT@set@ls@basefont 1697
 \MT@set@ls@font 1588, 1656, 1666
 \MT@set@named@keys 2827, 2841, 2863, 2882, 2896, 2904
 \MT@set@pr@codes 982
 \MT@set@pr@heirs 1055, 1228
 \MT@set@sp@codes 1380
 \MT@set@sp@heirs 1417, 1421
 \MT@set@tr@codes 1540, 1661, 1671
 \MT@set@tr@zero 1550, 1697, 3653
 \MT@Set@Expansion 2837, 2839
 \MT@Set@Extra@Kerning 2892, 2894
 \MT@Set@Extra@Spacing 2878, 2880
 \MT@Set@Protrusion 2823, 2825
 \MT@Set@Tracking 2859, 2861
 \MT@setup@ 656, 659, 3433, 3434, 4008
 \MT@setup@contexts 2440, 2455
 \MT@setupfont .. 808, 2376, 3669, 3672, 3676, 3710
 \MT@setupfont@hook 705, 729, 735, 740, 765, 772, 808
 \MT@shape 869, 968, 2070, 2082
 \MT@shorthand@off 3960, 3991, 3994
 \MT@show@pdf@annot 105
 \MT@shrink 259, 1289,
 3741, 3742, 3746, 3747, 3750, 3754, 3806, 3819
 \MT@shrink@ 1289, 1300, 1342
 \MT@shrink@default 275
 \MT@size 575, 579, 580, 869, 968
 \MT@size@name 568, 2097
 \MT@sp@cc@name 1387, 1389, 1441, 2880
 \MT@sp@context 2451, 2488, 4016
 \MT@sp@doc@contexts 2488
 \MT@sp@factor 250
 \MT@sp@factor@ 1134
 \MT@sp@inh@name 1415–1417
 \MT@sp@max 262
 \MT@sp@min 262
 \MT@sp@set@name 2708
 \MT@sp@split@val 1393
 \MT@sp@unit 256
 \MT@sp@unit@ 1142, 1444
 \MT@spacing 830, 1378, 3878
 \MT@spacing@false 246
 \MT@spacing@true 246
 \MT@split@codes 1024, 1028
 \MT@split@name 812, 869
 \MT@step 259, 1290,
 3744, 3745, 3761, 3768, 3820, 3823, 3824, 3828
 \MT@step@ 1290, 1300, 1342
 \MT@step@default 275
 \MT@stretch 259, 1288, 3738,
 3739, 3742, 3746, 3748, 3753, 3756, 3805, 3819
 \MT@stretch@ 1288, 1300, 1342
 \MT@stretch@default 275, 3739
 \MT@temp 1009–1012, 1431–
 1435, 1502–1505, 1728, 1732, 1735, 1740,
 1743, 1748, 1804, 1807, 1813, 1817, 1820,
 1823, 1828, 1835, 1840, 1902, 1905, 1922,
 2657, 2682, 3682, 3687, 3692, 3693, 3695, 3698
 \MT@temp@encoding 3222, 3223, 3233, 3239
 \MT@temp@family 3234
 \MT@temp@series 3235
 \MT@temp@shape 3236
 \MT@temp@size 3190, 3237, 3246, 3248, 3252
 \MT@test@east 1756, 1910, 2545, 2550
 \MT@text@ls 49, 1865, 1866, 1868
 \MT@the@pr@code 1016, 1042, 1048, 1586
 \MT@the@pr@code@tr 1016, 1586
 \MT@tlist@break 530, 565, 587, 2059, 3328
 \MT@toks 238, 1098, 1128, 1338,
 2155, 2183, 2190, 2237, 2267, 2286, 2298, 2306
 \MT@tr@cc@name
 ... 1631–1634, 1638, 1652, 1653, 2864, 2871
 \MT@tr@context 2451, 2488
 \MT@tr@doc@contexts 2488
 \MT@tr@factor@ 1640
 \MT@tr@font@list 1526
 \MT@tr@ispace 1561, 1562, 1646
 \MT@tr@ligatures 1577, 1649, 1709, 1712
 \MT@tr@max 262, 1889–1891
 \MT@tr@min 262, 1893–1895
 \MT@tr@noligatures 1577, 1706
 \MT@tr@okern 1575, 1648
 \MT@tr@ospace 1564, 1565, 1646
 \MT@tr@outer@icr 1817, 1842
 \MT@tr@outer@icr@ 1842
 \MT@tr@outer@l 1595, 1772
 \MT@tr@outer@next 1800, 1844
 \MT@tr@outer@r 1627, 1628, 1800, 1842
 \MT@tr@outer@r@ 1800
 \MT@tr@set@okern 1576, 1899
 \MT@tr@set@okern@ 1903, 1904, 1909
 \MT@tr@set@space 1567, 1724

- `\MT@tr@set@space@` 1729–1731, [1737](#)
`\MT@tr@set@space@@` 1742, 1747, 1750, [1755](#)
`\MT@tr@setname` [2708](#)
`\MT@tr@unit@` [1633](#), 1757, 1911
`\MT@tracking` 819, [1526](#), 1661, 2378, 3653, 3654, 3867
`\MT@tracking@` [1526](#), 3654
`\MT@trackingfalse` [248](#)
`\MT@trackingtrue` [248](#)
`\MT@try@order` [2054](#), [2062](#)
`\MT@undefined@char` [2230](#), [2232](#), [2242](#)
`\MT@upper` [2567](#), [2573](#), [2931](#)
`\MT@use@set` [2695](#), [2701](#), [2708](#)
`\MT@val` [2468](#)–[2470](#), [2472](#), [2478](#), [2480](#),
[2483](#), [2533](#), [2537](#), [2545](#), [2547](#), [2562](#)–[2564](#),
[2576](#), [2579](#), [2581](#), [2584](#), [2586](#), [2587](#), [2593](#),
[2595](#), [2596](#), [2598](#), [2601](#), [2603](#), [2614](#), [2617](#),
[2621](#), [2624](#)–[2626](#), [2628](#), [2630](#), [2637](#)–[2639](#),
[2642](#), [2651](#)–[2653](#), [2665](#), [2667](#), [2672](#), [2674](#),
[2676](#), [2678](#), [2917](#), [2919](#), [2927](#)–[2929](#), [2939](#)–
[2941](#), [2944](#), [2949](#), [3143](#), [3144](#), [3146](#), [3150](#),
[3171](#), [3177](#), [3181](#), [3182](#), [3347](#), [3348](#), [3351](#),
[3359](#), [3360](#), [3363](#), [3365](#), [3368](#), [3371](#), [3390](#),
[3391](#), [3393](#), [3395](#), [3573](#), [3574](#), [3576](#), [3583](#),
[3587](#), [3594](#), [3600](#), [3630](#), [3631](#), [3633](#), [3639](#), [3642](#)
`\MT@variants` [2047](#), [2749](#), [2759](#)
`\MT@vinfo` [75](#), [92](#), [825](#), [897](#), [1137](#),
[1146](#), [1150](#), [1185](#), [1346](#), [1363](#), [1542](#), [1552](#),
[1949](#), [1958](#), [1970](#), [1994](#), [2004](#), [2007](#), [2110](#),
[2112](#), [2118](#), [2449](#), [2472](#), [2781](#), [2787](#), [3474](#),
[3481](#), [3579](#), [3585](#), [3591](#), [3598](#), [3636](#), [3641](#), [3953](#)
`\MT@warn@ascii` [2192](#), [2285](#)
`\MT@warn@axis@empty` [2664](#), [2675](#), [2680](#)
`\MT@warn@code@too@large` [1112](#), [1115](#), [1119](#)
`\MT@warn@err` [75](#), [3478](#), [3479](#)
`\MT@warn@ex@too@large` [1322](#), [1325](#), [1336](#)
`\MT@warn@lua` [3848](#), [3860](#), [3871](#), [3882](#)
`\MT@warn@maybe@inputenc` ... [1944](#), [2299](#), [2307](#), [2310](#)
`\MT@warn@nodim` [3035](#), [3045](#), [3053](#), [3065](#)
`\MT@warn@number@too@large` [2219](#), [2291](#)
`\MT@warn@preset@t@width` ... [1241](#), [1261](#), [1445](#), [1515](#)
`\MT@warn@rest` [2172](#), [2296](#)
`\MT@warn@tracking@DVI` [1553](#), [3894](#)
`\MT@warn@unknown` [2169](#), [2304](#)
`\MT@warning` [66](#), [75](#), [93](#), [1890](#),
[1894](#), [1952](#), [2121](#), [2522](#), [2628](#), [2681](#), [2767](#),
[2789](#), [2872](#), [2973](#), [3036](#), [3076](#), [3089](#), [3151](#),
[3240](#), [3258](#), [3325](#), [3478](#), [3483](#), [3563](#), [3661](#)–[3663](#)
`\MT@warning@nl` [75](#),
[87](#), [94](#), [201](#), [222](#), [653](#), [687](#), [696](#), [742](#), [997](#),
[1090](#), [1126](#), [1178](#), [1218](#), [1262](#), [1337](#), [1717](#),
[2286](#), [2292](#), [2297](#), [2305](#), [2349](#), [2844](#), [3022](#),
[3335](#), [3341](#), [3351](#), [3479](#), [3484](#), [3516](#), [3544](#),
[3558](#), [3707](#), [3763](#), [3807](#), [3826](#), [3896](#), [3998](#), [4017](#)
`\MT@while@num` [596](#), [602](#), [768](#)–[770](#)
`\MT@with@babel@and@T`
... [663](#), [707](#)–[709](#), [731](#)–[733](#), [3986](#)–[3990](#), [3993](#)
`\MT@with@package@T` [661](#), [711](#), [714](#), [715](#),
[721](#), [728](#), [738](#), [750](#), [764](#), [766](#), [779](#), [790](#), [2360](#)
`\MT@xadd` [103](#), [498](#), [1995](#), [2000](#), [2005](#),
[2008](#), [2415](#), [2423](#), [2480](#), [2565](#), [2783](#), [2930](#), [3177](#)
`\MT@xaddb` [505](#), [2947](#), [3250](#)
`\MT@xdef@n` ... [322](#), [2711](#), [2714](#), [2741](#), [2746](#), [2770](#),
[2871](#), [2942](#), [2964](#), [3101](#), [3135](#), [3267](#), [3348](#), [3350](#)
- N**
- `\normalfont` [807](#), [3836](#), [3840](#)
`\nullfont` [807](#)
- P**
- `\pdfadjustinterwordglue` [3655](#), [3873](#)
`\pdfadjustspacing` [3624](#), [3816](#), [3839](#)
`\pdfappendkern` [3657](#), [3885](#)
`\pdfcopyfont` [849](#)
`pdfcpot (package)` [4](#), [27](#), [29](#), [51](#), [59](#), [89](#), [185](#)
`\pdffontexpand` [1300](#)
`\pdfnoligatures` [1951](#), [1957](#)
`\pdfprependkern` [3656](#), [3884](#)
`\pdfprotrudechars` [3623](#), [3727](#), [3864](#), [3905](#)
`pdfxcmds (package)` [36](#), [191](#)
`\pdftracingfonts` [298](#)
`\pickup@font` [753](#), [794](#), [796](#),
[799](#), [801](#), [2348](#), [2350](#), [2359](#), [2382](#), [2388](#), [2390](#)
`pifont (package)` [89](#)
`pinyin (package)` [52](#), [190](#), [191](#)
`protrusion (option)` [6](#), [7](#), [11](#), [113](#)
`PSNFSS (package)` [181](#)
`pstricks (package)` [9](#), [121](#)
`pxfonts (package)` [21](#), [130](#)
- Q**
- `qfonts (package)` [21](#), [130](#), [188](#)
- R**
- `ragged2e (package)` [75](#), [191](#)
`relsize (package)` [97](#)
`\rightmarginkern` [680](#)
`\rprcode` [1011](#), [1048](#), [1049](#), [1230](#), [1233](#), [1585](#)
- S**
- `selected (option)` [8](#), [14](#), [32](#), [64](#), [83](#), [114](#), [186](#)
`\selectfont` [1662](#), [2443](#), [2453](#), [3568](#), [4004](#)
`\set@fontsize` [2621](#)
`\SetExpansion` [14](#), [36](#), [2835](#)
`\SetExtraKerning` [18](#), [38](#), [2890](#)
`\SetExtraSpacing` [19](#), [39](#), [2876](#)
`\SetProtrusion` [13](#), [35](#), [2821](#)
`\SetTracking` [16](#), [37](#), [2857](#)
`\sfcode` [4015](#)
`\shbscode` [1412](#), [1413](#), [1424](#), [1427](#), [1434](#)
`\showhyphens` [3834](#), [3838](#)
`shrink (option)` ... [8](#), [16](#), [32](#), [116](#), [122](#), [123](#), [188](#), [190](#)
`slantsc (package)` [159](#)
`soul (package)` [4](#), [29](#), [52](#), [76](#), [189](#), [191](#)
`soulutf8 (package)` [29](#), [191](#)
`\spacefactor` [1783](#)
`\spaceskip` [1784](#), [1787](#)
`spacing (option)` [7](#), [11](#), [26](#), [30](#), [113](#)
`\stbscode` [1405](#), [1406](#), [1423](#), [1427](#), [1433](#)
`step (option)` ... [8](#), [30](#), [32](#), [116](#), [122](#), [123](#), [186](#), [188](#), [190](#)
`stretch (option)` ... [8](#), [16](#), [25](#), [32](#), [116](#), [122](#), [123](#), [190](#)
- T**
- `\tagcode` [1581](#), [1943](#), [1948](#)
`tex4ht (package)` [50](#), [189](#)
`\textls` [23](#),
[52](#), [53](#), [758](#), [781](#), [786](#), [1598](#), [1659](#), [1674](#), [1864](#)
`\textls*` [23](#)
`\textmicrotypecontext` [22](#), [48](#), [754](#), [2447](#)

tikz (package)	72, 192		
tipa (package)	89		
trace (package)	90, 189		
\tracingmicrotype	86		
\tracingmicrotypeinpdf	97		
\tracingmicrotypeinpdfallfalse	104		
\tracingmicrotypeinpdfalltrue	104		
tracking (option)	7, 11, 23, 30, 113		
txfonts (package)	21, 130		
U			
ucs (package)	86, 191		
ulgothic (package)	21		
unit (option)	8, 31, 187–189		
\UseMicrotypeSet	11, 33, 60, 72, 2516, 2692		
V			
verbose (option)	9, 30, 31, 115, 186, 187, 191		
W			
wordcount (package)	49, 190		
X			
\xspaceskip	1786, 1789		
Y			
yfonts (package)	24		
Z			
zefonts (package)	21, 129		

C The L^AT_EX Project Public License

LPPL Version 1.3c 2006-05-20

Copyright 1999, 2002–2006 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 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

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.
2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.
5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
 - (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
 - (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.
- (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
- (d) You distribute at least one of the following with the Derived Work:
 - i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
 - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.
10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
 - (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions

in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work

by any means.

12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any au-

thor named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
 - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
 - (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current

Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a L^AT_EX work, this could be done, for example, by posting to `comp.text.tex`.)

3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.
- (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.
4. If you make an ‘intention announcement’ as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.
5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document ‘modguide.tex’ in the base L^AT_EX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing L^AT_EX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L^AT_EX, the discussion in ‘modguide.tex’ may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
% http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status ‘maintained’.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the ‘Work’ referring to the three files ‘pig.dtx’, ‘pig.ins’, and ‘pig.sty’ (the last being generated from ‘pig.dtx’ using ‘pig.ins’), the ‘Base Interpreter’ referring to any ‘L^AT_EX-Format’, and both ‘Copyright Holder’ and ‘Current Maintainer’ referring to the person ‘M. Y. Name’.

If you do not want the Maintenance section of LPPL to apply to your Work, change ‘maintained’ above into ‘author-maintained’. However, we recommend that you use ‘maintained’ as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case (e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

Important Recommendations

Defining What Constitutes the Work

The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.